State of the art

March 24, 2020

The exploration of an unknown environment pursued by a team of robots is a complex problem, tackled in different ways across the literature. It can be informally defined as the process of producing a representation of the environment, in the following referred to as map, which can be used for future navigation. The map can fall into two categories; it is said to be topological if it is a graph modeling connectivity between regions, while it is metric if it provides the exact locations of obstacles. It is useful to define occupancy grids and coverage maps, being two of the most used metric models for the map. *Occupancy grids* model the environment as a grid where each cell can be marked as *free* or *obstacle* if already scanned through sensors, or *unknown* if it has not been scanned yet. Also, *coverage maps* are a grid-based representation of the environment, where each cell contains the posterior probability of being covered by an obstacle. This provides different advantages compared to occupancy grids, like for example the possibility to finely model a wall not parallel to x- or y-axis of the grid, without the need of enlarging it to match the discretization.

The mapping process is carried on by one or more robots able to perceive the environment utilizing sensors of different types. The most common are lasers [22, 11] and sonars [31], even if other types of sensors are sometimes used, like a laser-limited sonar [8, 9] which is a combination of the two, or a Microsoft Kinect sensor [25], which provides 3D measurements based on an RGB camera and a depth sensor. The map is progressively updated by including the information obtained from sensors on the partial map known at that moment.

Looking at exploration from a conceptual point of view, two main phases can be identified. The first one concerns the detection of the best locations to explore next in the partial map built so far. The other one deals with the allocation of robots to these candidate locations. Thus, the whole exploration can be seen as an iterative two-step procedure where, once the map is updated, a set of possible points of interest are chosen, based on some criterion, and then to each robot is assigned a goal location. These two steps are repeated until the exploration can be considered finished. The *exploration strategy* is the algorithm that selects the candidate locations, while the *coordination mechanism* is the one allocating the robots to them.

1 Exploration strategy

As previously described, the exploration strategy is an algorithm providing the candidate locations robots should visit to maximize their knowledge about the environment. There are two crucial aspects in this, that are the definition of a candidate location and the criterion used to choose the best one.

How a candidate location is defined depends strongly on the representation of the environment. As presented in the previous section, this can be divided into two categories, topological or metric, which turns into graph-based or grid-based representations, even if other data structures are possible, like in [1] where the map is stored as two lists of line segments.

The first step of the exploration strategy consists in generating a set of candidate locations. Its generation has a high impact on the definition of the algorithm and it is the core in frontier-based strategies [8, 9, 10, 11], where the focus is mostly in this generating step, rather than in the choice of the next location among the possible candidates. Other strategies, on the contrary, skip the generation of this set by considering the whole set of known cells or a portion of them. This is possible because of the particular implementation of the criterion used to choose the next best location [5].

The choice of the next location to explore among the set of candidate ones is done in different ways across the literature. An example of this in a metric representation is in [1], where a comparative review of four strategies for single robot exploration is performed, distinguishing among a random approach (used as benchmark), a greedy one and two complex *ad hoc* procedures, testing their performance over different environments. Therefore, the choice can be done according to different criteria and the two main factors affecting it are the expected utility provided by the location and the cost of reaching it. In a topological representation, strategies like a Depth-First Search [2] and Breadth-First Search [3] are naturally possible algorithms on undirected graphs, while [4] proposes an interesting approach to solve the exploration problem on directed graphs.

In the following, some exploration strategies are presented, distincted into three categories: *information gain-based, frontier-based* and *topological strategies*. We give particular attention to the second one, being the one used in this work.

1.1 Information gain-based strategies

Information gain-based strategies as presented in [5] are probabilistic strategies usually employed on coverage maps, and consequently extensible to occupancy grids. Candidate locations are chosen among the known cells of the grid, according to the expected change in the entropy obtainable by moving a robot there. Given a posterior probability distribution *p(x)* of a cell *x*, its entropy *H(p(x))* is defined as

$$H\left(p\left(x\right)\right) = -\int_{x} p\left(x\right) \log p\left(x\right) dx$$

While, the information gain for a given cell c and measurement z taken from the pose x is

$$I(c,x,z) = H(p(c)) - H(p(c|x,z))$$

Then, each known cell in the grid is considered as a possible candidate location and the one providing the highest expected entropy reduction, i.e., information gain, is selected. This method provides suitable locations because the information gain for a completely known cell is near zero, thus the approach tends to assign candidate locations in the proximity of uncertain cells, increasing the knowledge of the environment. This partially justifies the absence of care in generating the set of candidate locations, rather it is preferred to take the whole set of known cells and to check the information gain each one can provide.

The impact on the performance of this brute-force search is high, for this reason, in [5], besides the basic strategy presented so far, two modifications are also introduced. The first one reduces the number of candidate locations from the whole map to the ones in a local window, which has to be completely explored before moving on. Rather than reducing the set of candidate locations, the second one modifies the way in which the next one to explore is chosen by introducing the cost to reach that location from the robot pose.

In [6], it is presented the *A-C-G strategy* which defines a conceptually similar approach but takes explicitly into account the contribution to entropy of the points sensed from the candidate location, discriminating between already sensed points and the ones sensed for the first time. It also includes a factor proportional to the distance from the robot location and the candidate one.

In [7], an information gain strategy is integrated into the localization and mapping phase. This allows deciding which action to perform at each step of the exploration, by taking into account the trajectory and map uncertainty.

1.2 Frontier-based strategies

As presented above, information gain-based strategies mainly focus on the process of deciding the next location to explore, giving less attention to the definition of the set of candidate locations. This is extremely clear by considering the base method, where all the cells in the grid are possible candidate locations and the method implicitly cuts out the ones not providing new knowledge. In frontier-based strategies, the focus is shifted to the creation of a good set of candidate locations.

As defined in [8], the paper originally introducing this strategy for single robot exploration, a *frontier* is the boundary region between explored and unexplored space. The idea is that, by assigning a robot to its closest frontier as location to explore, the line between explored and unexplored space is pushed continuously, until the whole environment is mapped. In that case, occupancy grids are used to model the environment and with that representation, it is pretty straightforward to find out a frontier, identifiable with a cluster of adjacent free cells which neighbors are unknown.

This simple idea works extremely well in practice and for this reason, it has been used widely in the literature, producing a lot of extensions and adaptations to the various cases, like [9] where the strategy is extended to multi-robot scenarios.

In [10], the Leader-Follower exploration algorithm is presented. It focuses on the roles assumed by the robots during the exploration, which can be dynamically changed, according to the distance from the assigned location. Candidate locations detection is done by identifying frontiers and, differently from the strategy of [8] and [9], the next to explore is not chosen as the nearest one. Indeed, it looks for the pair of frontiers maximizing the sum of the rewards for the leader and the follower, where the reward function is composed of a utility term minus the cost to reach the frontier.

[11] provides an extension of this exploration strategy to topological maps, rather than occupancy grids. An interesting aspect of this is in the strict relation between the frontiers and the nodes of the graph. As the environment is progressively mapped, frontiers are detected and classified according to geometric information about the environment into free area or transit area, defined as the area where the robot transits between two spaces (rooms, corridors, and so on). Once classified, in one of the two categories stated above, the next frontier to explore is selected through a cost-utility function, composed of three terms: the geometric and the semantic utility, and the topological cost. The geometric utility corresponds to the size of the frontier; a bigger frontier offers a bigger range to acquire new information. The semantic utility is related to the classification of the frontier, being a transit area preferable over a free area, despite its smaller size. The topological cost is a cost term associated with the connectivity between frontiers. It assigns a fixed small cost to consecutive frontiers, while if to reach a frontier, a robot has to pass by other frontiers, the cost of that one is proportional to the number of crossed frontiers. Once a frontier has been explored, it is added as a node in the graph. The proposed cost-utility function is then linearly related to the utilities and the relation with the cost factor is a reverse exponential. The algorithm guided by this function is shown to have good performance both in terms of exploration time and traveled distance against some benchmark algorithms.

1.3 Topological strategies

Topological strategies rely on a graph-based representation of the world. This is useful to neglect the geometrical features of the environment and to focus on its structure. As shown in [12], the complexity of using geometric maps grows exponentially as the environment becomes larger and this justifies the use of topological maps. Moreover, the use of a directed graph can also simulate the case of one-way streets, where the robot is allowed to go in one direction and not in the opposite one [13], which would be impossible to describe just relying on geometric maps.

In this kind of strategies, candidate locations are nodes of the graph and the next one to explore is decided in various ways, strongly depending on the type of graph used. In fact, in the literature, both undirected and directed graphs are used, with a further distinction whether or not the vertices are identifiable. A vertex is identifiable if it can be recognized by a robot when revisited. This is not always guaranteed because the robot may have limited sensor capabilities or the appearance of vertices may change.

Undirected graphs with distinguishable vertices are the most straightforward case. Each vertex is labeled uniquely and the robot is allowed to traverse the edges in both ways [2, 14]. In [15] it is presented an extension of the Depth-First Search algorithm to the multi-robot scenario both for graphs and trees. A particular aspect of the model used is that edges are considered as opaque, this means that from either end, it is not clear where the edge goes. In [16] it is analyzed the problem of piecemeal exploration, this states that the robot can traverse a limited number of edges before going back to the source vertex. It is a realistic context in which the robot has a limited amount of fuel or battery and needs to refill it after a fixed number of steps or traveled distance. The algorithm proposed for this problem is based on Breadth-First Search and another important aspect is the use of the concept of *frontier vertex*, defined as a vertex incident to unexplored edges.

Undirected graphs with anonymous vertices introduce some difficulties and to get rid of them, markers are needed to distinguish between explored and unexplored area [3, 17, 18]. In [17] it is shown that one marker is sufficient to allow the robot to build a graph isomorphic to the environment in low-order polynomial time and the use of multiple markers may improve the performance. In [3] two enhancements are presented both to single-robot and multi-robot exploration in such environments, provided by the use of a Breadth-First Search and the exploitation of local neighbors information.

In the case of directed graphs, the robot movement is strongly limited with respect to undirected graphs. Clearly, Depth-First Search is not always possible because backtracking is not guaranteed to be applicable. Different algorithms have been proposed to deal with these models [4, 13, 19]. In [4] it is proposed an algorithm to visit all nodes and edges with a subexponential upper bound on the number of edge traversals. In [13] it is defined an algorithm able to explore a directed graph with anonymous vertices by using two robots through the simultaneous learning of the graph and a homing sequence. This is done by keeping multiple possible maps, updating them through a sequence of movements, then checking their correctness. It also states that it is not possible to efficiently learn the same kind of graph utilizing a single robot with a constant number of pebbles without prior knowledge on the number of vertices. In this case, pebbles are used similarly to the markers stated above. They can be dropped by a robot at a certain vertex to make it recognizable when revisited and eventually, they can be also picked up by the robot to place them at another node. Previously it has been presented [17], which shows that same problem solvable with one marker in low-order polynomial time in the case of an undirected graph.

2 Coordination mechanisms

In a multi-robot scenario, once candidate locations are detected on the map, it comes out the problem of how the robots in the team have to be assigned to them in order to maximize the knowledge about the environment. Moreover, even if a random allocation is possible, it is clearly preferable an assignment of agents to candidate locations which minimizes some metrics like the time taken to explore the environment or the distance traveled by the robots.

The answer to this is provided by the coordination mechanism, which is the algorithm that assigns robots to candidate locations according to some criteria. Coordination mechanisms are distinguished into online and offline. Online mechanisms assign robots to candidate locations by taking into account the actions currently done by the other members of the team [20, 21, 22]. In offline mechanisms, in contrast, roles are assigned to robots before the exploration starts and offline coordination can be divided into two further categories, fixed and variable [31]. In fixed offline coordination, robots act according to the roles defined before the beginning of the exploration and they stick to these roles, without altering them [23, 24, 25, 26]. In variable offline coordination, robots can exchange their roles dynamically as the exploration goes on [10].

Coordination based on an online mechanism is weighed down by the need for more communication among the agents. Before an allocation is made, an agent has to know other agents poses and targets locations. On one hand, this implies a lot of communication to make proper assignments; on the other hand, this allows to perform choices aimed at maximizing the performance of the system. To clarify this, it is interesting to anticipate the algorithm proposed in [20] and analyzed more in-depth in the following section. This algorithm provides that every time the map is updated and the set of candidate frontiers is detected, each robot communicates its expected gain obtainable from the exploration of each frontier in the set. After having received them all, the central executive computes the next location for each robot, in a way to provide the highest possible gains for the whole system.

Through the use of roles, offline mechanisms require little to no communication once the exploration is started, making the robots and the whole system easier to implement. The other side of the coin is that robots move almost freely, with the possibility of interference among them and the redundancy of assignments to the same target location.

The relation between exploration strategy and coordination mechanism is quite tight and the relative impact each one has on the performance is hard to establish. A work in this sense is [27], where different exploration strategies and coordination mechanisms are compared in two different environments. What comes out is that in structured environments, like an office one with a lot of rooms and corridors, the detection of good candidate locations is preferable over a good assignment of robots to them. In an open environment, the contrary holds, being the coordination mechanism able to increase the amount of area explored in the same amount of time, making the impact of the exploration strategy less relevant.

2.1 Online mechanisms

Online coordination mechanisms allocate robots to target locations exploiting current information about other robots actions. To achieve this, robots need to communicate with each other. This has been done in different ways across literature, using different techniques [20, 21, 22].

In [20] the communication is performed through the use of *bids*. Every time a robot receives a map update from the central mapper, it sends a bid with a list of costs and information gains for each frontier to the central executive. As the central executive gets all the bids, computes the assignment maximizing the difference between information gain and cost for any robot and assigns the frontier to that robot. Once an assignment is fixed, the other bids are discounted by a certain value to take it into account. This procedure is iterated as long as there are no remaining robots or tasks. The discount factor is fundamental, being the main factor introducing the online coordination aspect of this algorithm, in fact, if bids were not discounted, each robot would go towards the frontier with the highest estimated utility, not taking into account other robots assignments. Also [21] and [22] perform coordination by considering the utility of each frontier computed as the difference between the information gain it can provide and the cost to reach it.

In [21] every time a robot is assigned to a certain frontier, the utility of the other frontiers is discounted by a value proportional to the probability of being in the visibility area from the assigned one. In [22] this approach is extended to limited communication scenarios.

[28] differs from the previous works because the algorithm proposed uses a topological map, rather than a metric one. The topological map is built as the Voronoi graph of the partial map, known up to that moment. A Voronoi graph is a graph in which nodes consist of points of the free space equidistant from the closest obstacles. An edge connects a pair of nodes if they are adjacent in the map. Once the Voronoi graph is computed, it is segmented in a way to create frontiers at *critical points,* like doorways. At this point, for each robot is computed the cost for reaching each map segment and the optimal allocation is then found by applying the Hungarian method, which is an algorithm able to provide the optimal solution with minimal cost.

2.2 Offline mechanisms

Offline coordination mechanisms provide a definition of roles prior to the beginning of the exploration. By sticking to these roles, coordination among robots needs little to no communication, which is one of the main advantages of this approach. Roles definition may also be modified at run-time, like how is done in the Leader-Follower algorithm [10] presented above, where the role depends on the distance from the assigned frontier.

Two major works following this approach for this thesis are represented by [25] and the further extension provided by [29]. In [25] three coordination mechanisms, namely *reserve, buddy system,* and *divide and conquer* are presented. The *reserve* mechanism splits the team into two smaller teams where one is left idle at the initial position, while the second one is sent to explore frontiers. As new frontiers are found, idle robots are progressively turned into active agents and assigned to them. Once all the initially idle robots are active agents, the exploration is carried out without further coordination. *Buddy system* works in a similar way, with the difference that rather than considering single robots, pairs of robots are considered. At the start of the exploration, some pairs are sent to explore frontiers, while the others remain idle at the starting position. Once a branching point is found, that is a zone of the environment where different spaces met, like a T-shaped junction, for example, the pair is split and each robot explores a different branch. If another branching point is found, one branch is explored by the single robot which discovered it, and the other one is assigned to a pair from the idle set, which then turns into active. In *divide and conquer*, at the beginning of the exploration, all the robots move together following a leader, then as a branching point is found, the team splits into two halves. A new leader for the second team is decided and each team is assigned to a branch. This splitting approach goes on while there are teams composed of more than one robot and, after that, they proceed in an uncoordinated way.

[29] modified these mechanisms proposing respectively *proactive reserve, proactive buddy system,* and *side follower.* The idea behind the first two mechanisms is to move the idle set from the starting position, towards a better position, nearer to the possible branching points. This would allow the robots turned into active to reach the target positions in less time, once they are called. The waiting position for the idle team is computed as the barycenter of the locations of the active agents. This thesis expands this approach modifying the way in which this waiting position is computed by taking the topology of the environment into account. Differently from *divide and conquer,* the *side follower* mechanism organizes the agents into groups of three, rather than a single group. The idea is that each robot in the group has a preferred direction for the frontier to explore: the left robot tends to explore frontiers on the left, the right robot prefers the ones on the right and the robot in the center explores the ones in front of it. These modifications are shown to have very good performance when compared with the benchmark ones, particularly *proactive reserve* which is usually better than all the other considered strategies. *Proactive buddy system* outperforms the simple *buddy system* mostly on open environments, resulting in similar or worst performance on the others. *Side follower* also performs generally better than *divide and conquer*, particularly on the environments reflecting the structure for which it is designed, that is a central corridor with spaces on the sides.

References

[1] F. Amigoni. Experimental Evaluation of Some Exploration Strategies for Mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2818–2823, 2008.

[2] A. Dessmark and A. Pelc. Optimal graph exploration without good maps. In Proceedings of the European Symposium on Algorithms (ESA), pages 374–386, 2002.

[3] H.Wang and P. Dymond. Enhancing exploration in graphlike worlds. In Proceedings of the Canadian Conference on Computer and Robot Vision, pages 53–60, 2008.

[4] S. Albers and M. R. Henzinger. Exploring unknown environments. SIAM Journal on Computing, 29(4):1164–1188, 2000.

[5] C. Stachniss. Robotic Mapping and Exploration. Springer, 2009.

[6] F. Amigoni, V. Caglioti, and U. Galtarossa. A mobile robot mapping system with an information-based exploration strategy. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics, pages 71-78, 2004.

[7] C. Stachniss, G. Grisetti, W. Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In Proceedings of the Robotics Science and Systems I, pages 65-72, 2005.

[8] B. Yamauchi. A frontier-based approach for autonomous exploration. In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997.

[9] B. Yamauchi. Frontier-based exploration using multiple robots. In Proceedings of the Second International conference on autonomous agents (Minneapolis 1998), pages 47–53, 1998.

[10] B. Wang and S. Qin, Multi-robot Environment Exploration Based on Label Maps Building via Recognition of Frontiers. In proceedings of IEEE International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014.

[11] C. Gomez, AC. Hernandez, R. Barber. Topological Frontier-Based Exploration and Map-Building Using Semantic Information. Sensors. 19(20):4595, 2019.

[12] R. Chatila, J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Volume 2, pp. 138–145, 1985.

[13] M. A. Bender and D. K. Slonim. The power of team exploration: Two robots can learn unlabeled directed graphs. pages 75–85, 1994.

[14] L. Gasieniec P. Fraigniaud and D. Kowalski. Collective tree exploration. Networks, 48:166–177, 2006.

[15] J. Xiao P. Brass, A. Gasparri. Multirobot Tree and Graph Exploration. IEEE Transactions on Robotics, 27(4):707–716, 2011.

[16] M. Betke B. Awerbuch and R. Rivest. Piecemeal graph exploration by a mobile robot. In Proceedings of the 8th Annual ACM Conference on Computational Learning Theory (COLT), pages 374–386, 1995.

[17] E. Milos G. Dudek, M. Jenkin. Robotic Exploration as Graph Construction. IEEE Transactions on Robotics, 7(6):859–865, 1991.

[18] F. Hoffmann. One pebble does not suffice to search plane labyrinths. Fundamentals of Computation Theory, 117:433–444, 1981.

[19] C. Papadimitriou X. Deng. Exploring an Unknown Graph. In Proceedings of the IEEE 33th Annual Symposium on Foundations of Computer Science, pages 355–361, 1990.

[20] W. Burgard, R. Simmons. Coordination for multi-robot exploration and mapping. In Proceedings of the National Conference on Artificial In- telligence (2000), volume 4, pages 7–27, 2000

[21] M. Moors W. Burgard and D. Fox. Collaborative multi-robot exploration. In Proceedings of the IEEE International Conference on Robotics and Automation (2000), pages 476–481, 2000.

[22] M. Moors W. Burgard and M. Schneider. Coordinated multi-robot exploration. IEEE Transactions on Robotics, 21(3):376–378, 2005.

[23] J. Ko D. Fox and K. Konolige. Distributed multirobot exploration and mapping. IEEE special issue on multi-robot systems, 94(7):1325–1339, 2006.

[24] J. Stromand R. Morton E. Olson. Progress towards multi-robot reconnaissance and the magic 2010 competition. Journal of Field Robotics, 29(5):762–792, 2012.

[25] C. Nieto-Granda H. Christensesn and J.G. Rogers III. Coordination strategies for multi-robot exploration and mapping. The International Journal of Robotics Research, 33(4):519–533, 2014.

[26] D. Fox R. Vincent and J. Kol. Distributed multirobot exploration, mapping, and task allocation. Annals of Mathematics and Artificial Intelligence, 52(2):229–255, 2008.

[27] N. Basilico F. Amigoni and A.Q. Li. How much worth is coordination of mobile robots for exploration in search and rescue? RoboCup 2012: Robot Soccer World Cup XVI, 2013.

[28] K. Wurm, C. Stacniss, W. Burgard. Coordinated Multi-Robot Exploration using a Segmentation of the Environment. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1160-1165, 2008.

[29] M. Cattaneo. An Analysis of Coordination Mechanisms for Multi-Robot Exploration of Indoor Environments. 2017. Master Thesis, Politecnico di Milano, Italy.

[30] E. Todt, G. Rausch, R. Suárez. Analysis and classification of multiple robot coordination methods. In Proceedings of the IEEE International Conference on Robotics and Automation. 4:3158 - 3163. 2000.

[31] A. Cheyer D. Guzzoni and L. Julia. Many robots make short work. Artificial Intelligence Magazine, 18(1):55–64, 1997.