Riassunto papers

June 1, 2018

1 “Collaborative Multi-Robot Exploration” - Burgard

In this paper is considered the problem of exploring an unknown environment by a team of robot. The main focus is on the coordination of a team of multiple robots by the definition of target points for the exploration.

The approach presented in this paper uses occupancy grid maps to represent the environment and assumes robots to know their relative position during the whole exploration process. This assumption is useful to allow a simple computation of the integrated occupancy grid map. The selection of target points to explore is based on two concepts: the cost of reaching it and the utility it provides. Cost for traversing a cell is proportional to the occupancy value of that cell and through that definition, the minimum cost path between two cells can be computed. The utility of a target point is calculated as the expected visible area from that point. To find out this value, it exploits an heuristic based on the observation that a robot exploring a big open terrain can cover much larger areas than a robot exploring a narrow part of the environment. This is achieved by counting the number of times $h(d_i)$ the distance d_i was measured by any of the robots. The value of $h(d_i)$ is then used to compute the probability that the robot's sensors covers objects at distance d. A great advantage of the algorithm introduced is that two robots never choose the same target point because once a robot is assigned to it, utility of that point is decreased accordingly.

The provided experimental results only deal with an office-like environment and show that the coordination always provides better performances. Moreover, two coordinated robots have performances similar to three uncoordinated robots and the improvement introduced by the coordination is almost the same both with two and three robots.

2 “Coordinated Multi-Robot Exploration” - Burgard

As in the previous paper, the focus of this one is on the coordination of a team of multiple robots by the definition of target points for the exploration, but it also takes into account limited communication among them.

The algorithm used for the assignment of the target points is almost the same as above. Few differences are introduced in the reduction of the utility of a frontier cell after a robot is assigned to an adjacent one and in determining the pair robot-frontier cell. In fact, in the previous case, cost and utility had the same weight, now the relative importance of utility versus cost is modified through a positive factor β. Moreover, experiments performed, showed that if $\beta \in [0.01, 50]$ the exploration time is almost constant. While if $\beta > 50$ the impact of the coordination decreases, thus the exploration time increases. While if it is near 0, robots ignore distance to be traveled, increasing again the exploration time.

Limited communication range affects highly the algorithm explained before, but it is easily modified to deal with this problem by applying it to every sub-team of robots which are able to communicate with each other. Another issue derived from the limited communication consists in when new assignments have to be computed. In fact, in the case of perfect communication, they can be calculated whenever a robot reaches its target. In the other case, this approach can't be pursued. Therefore, what is done is to store in each robot the latest target locations assigned to other robots; in this way, robots avoid going to places already explored by other robots.

It is meaningful to point out that to achieve coordination, the team must be able to communicate the maps of the individual robots during exploration. Main idea is to split the team of robots into clusters, and messages sent by a robot are forwarded to all team-mates in the corresponding cluster. If every cluster maintaines a map built from all observations made by robots of that team, if connection problems are encountered, it may happen that the map is updated twice, which is not admissible. An efficient way to overcome this problem is to have each robot storing a log of sensors measurements perceived by this robot and only trasmits those measurements that have not been transmitted to the corresponding robot so far. Additionally, a data structure containing the time stamp of the last transmission to each robot is stored. In this way, as a measurement has been sent to all other robots can be discarded.

The experiments are performed on three different maps: an unstructured one, an office-like and a corridor environment. In all the three cases, the coordinated algorithm outscales the uncoordinated one. This is due to the better distribution of the robots in the environment, allowing to speed up the exploration.

3 Coordination for Multi-Robot Exploration and Mapping - Simmons

This paper main topics are the definition of a mechanism for coordinated mapping and another for coordinated exploration.

The coordinated mapping approach works with (reasonably) static world and assumes that in the beginning, robots know their relative position. It decomposes the mapping problem in a modular way, being composed of a local map stored on each robot and a global map kept by a central mapper. As a robot receives its own odometry and sensor measurements, it computes a maximum likelihood extimation of its position, a maximum likelihood extimation for the map and a posterior density of its "true" location. This update takes into account both noise in motion and in perception, moreover it converges fast. For what concerns the central mapper, its main role is to integrate informations coming from the robots in real-time. This information are sent after having collected a certain number of them and the central mapper combines them by minimizing the error between the scans of the different robots.

The coordinated exploration approach makes each robot construct a "bid", on the basis of the expected utility for it to travel to a frontier cell and the expected information gain there. Cost is computed as the optimal path from the robot's current position. The information gain is estimated by assuming that the robot has a nominal sensor range and then counting the unexplored cells which fall within the radius of the frontier cell. Also the maximum and minimum extent of the information gain region are stored because they are used to forme a rectangle that approximates this region. It is useful to find potential overlaps in coverage. This approach makes robot move enough far to optimize exploration but if there is an obstacle separating them like a wall, they move near it but in the different rooms. The algorithm used to assign tasks to robots is a greedy one, assigning the first task to the bid with highest net utility. After that, other bids are discounted by estimating the percentage of overlap with the already assigned task and then it assigns new task with same criteria as before. This process goes on up to the end of the tasks or the available robots. Moreover, it is introduced the concept of hysteresis, which is useful to handle the high variation that might be measured in the information gain metric.

The experiments are performed on real robots in an hospital building composed of corridors and offices. Main behaviours to point out are that three robots start in a narrow corridor, then one will stand still and the other two will go to the frontiers and that if one robot goes near a task assigned to another one, then the task is dynamically swapped to the nearer robot.

Other reported experiments run in simulator have been performed in five different environments: a single-corridor office, a two parallel corridors office, an obstacle-free environment and two random unstructured environments. In all the cases, increasing the number of robots improves results but the amount of improvement depends on the characteristics of the environment. In fact, in the first two environments, moving from two to three robots doesn't have the same impact as switching from one robot to two because the third robot is almost useless. In the obstacle-free environment, a single robot exploration has an high rate of failure, which is highly mitigated by switching to two robots and it's even better with three. Also the accuracy of the map increases with the number of robots. For what concerns the unstructured environments, the presence of obstacles helps in spreading the robots around, so this makes the improvement of using three robots rather than two way higher than moving from one to two.

4 Coordination strategies for multi-robot exploration and mapping - Christensen

Aim of this paper is to evaluate different strategies for coordinating a team of robots during exploration of an unknown environment. The strategies introduced are three, namely *Reserves, Divide and Conquer* and *Buddy system.* Difference among them mainly consists in how the *proactive* extra members of the team are when all paths or frontiers are assigned to other members of the team. In *Reserves*, extra robots wait in the starting area until they are needed. In *Divide and Conquer*, robots travel as a group and split in half when new navigaton goals are uncovered. In *Buddy system*, robots travel in teams of two members and split as in *Divide and Conquer*.

The mapping system is centralized with a master map coordinator and an instance of the library plus some mapping plugins running on each robot. As each robot senses the surronding environment, sends the relevant information to the coordinator, summing up the relevant features measurements, its best odometric estimate of its pose and its posterior estimate of its position in its own local map frame. In this way, the communication overhead is minimized and the master map coordinator is able to merge the local maps into a global one, being able to compute the transformation between the global map frame and each robot's local map frame. Map coordinator needs an initial estimate of the relative pose of the robots in the team; this is to determine which measurements are of the same landmarks and merge them together.

Exploration strategy used is the frontier-based one and each coordination strategy defined uses a greedy approach consisting in minimizing the distance travelled to assign the navigation goal to a team. The different coordination strategies presented before aim at trading off *availability* and *non-interference.* First parameter explains the possibility for the robots to be close and explore branching structures quickly. The second one is the possiblity for the robots to not get in each other's way. In particular, the three strategies can be analyzed through this two parameters and what can be pointed out is that: *Reserve* has low availability and low interference; *Divide and Conquer* maximizes availability but potentially causing interferences between robots; *Buddy system* provides good availability without incrementing interference.

Simulation experiments are performed in three different scenarios: an home/office-like, a simple maze and a series of rooms with large obstacles connected by long corridors.

In the first one, due to the large size of the spaces, interference between robots is not a main problem and the availability should be maximized. This is also because of the high number of connections between rooms, which makes new frontiers to be uncovered quickly. Simulations runned show that the time taken to explore reduces as the number of robots increases up to nine, after which there is a plateau. *Divide and Conquer* outperforms other strategies for almost all robot team compositions and maintains its performance at larger team sizes better than *Reserves.*

The second environment is a simple maze with a low branching factor, which means that interference between robots might be an issue and the availability of additional robots would help but not as much as in the previous case. Simulations runned show that in this case additional robots are not needed beyond the number of exploration frontiers which are open at a given time, so using more than six robots is useless. *Buddy system* outperforms the other strategies and the additional availability provided by the *Divide and Conquer* strategy optimizes the use of small teams compared to the *Reserves* strategy.

In the last simulated environment, robots have to travel a long distance before branching points, so the availability should be helpful for exploring the map quickly, while a large team size might not help much. As expected, performance didn't improve beyond 7 robots and the best strategy is *Buddy system*, which shows that each robot encounters a single branch point by the time a new team can be re-formed.

Live experiments are performed in an office and in a training facility, consisting of several buildings simulating a small village. Performance measured the number of exploration goals explored within a certain amount of time. In the office, *Buddy system* was not tested because still not available.

The first live experiment was runned with two different starting locations. The first one was characterized by a very limited maneuver possibility, so interference among robots was very high; while the second one started from an area with more room to maneuver. With both starting locations, *Divide and Conquer* performed better than *Reserves* and an explanation to this is that the latter causes more robots to be making observations of exploration frontiers due to the fact that groups contain more than one robot. In this way, frontiers are found faster and more points-of-interest are explored. Moreover, with the second starting location, *Divide and Conquer* had less interference because the entire team moved together out of the starting area before any divide operation; this is not the case for the *Reserves* strategy, which had to maneuver robots out of the starting location.

Differently from the first live experiment, the second one used as metric for comparing strategies the amount of time elapsed to fully explore each building. In this case what is found out is that *Divide and Conquer* and *Buddy system* always perform better than *Reserves* and performances always increase with the team sizes.

5 Exploration strategies - Amigoni

A comparison between different exploration strategies for a single robot is done in this paper. To do this, it is considered an holonomic robot moving in a two dimensional environment. Sensing the world through a 360° sensor with a limited radius. This radius can assume different values and is used as a metric of comparison. Evaluation of the exploration strategies is performed by generating a set of random candidate observation positions along the boundary between known and unknown space. Reachability of such candidate observation position

p is checked and then each one is evaluated through an evaluation function $f(\cdot)$, different for each considered strategy. In the end, the best candidate observation position is selected as the one which maximizes $f(\cdot)$.

Four strategies are considered, random strategy, greedy, GB-L and A-C-G. In the first, $f(\cdot)$ is simply a random function, so the next observation position is selected randomly. The second one evaluates an obsevation position on the amount of new information the robot is expected to gain from reaching it. GB-L is an example of exploration strategy using an *ad hoc* evaluation function, in fact it is computed taking into account the amount of new area sensed in the observation position p and the lengths of the paths from the robot current position to p. A-C-G is an example of complex and theoretically found exploration strategy, using as $f(\cdot)$ derived through the concept of relative entropy. It is mainly composed of four terms, the contribution to the entropy of the points sensed from p, the contribution to the entropy of the points sensed for the first time from p, the contribution to the entropy of the already knwon points sensed again from p and the contribution to the entropy of the cost of reaching p.

Experiments have been runned in a simulator, using two different metrics of comparison: the number of sensing operations needed to complete the exploration and the total distance travelled by the robot during the exploration. Three different environments are explored: an office-like environment, a large open space and one scattered with several obastacles. Results pointed out interesting facts, like that the office environment is not alwasy mapped faster than the large obstacle environment, whose mapping is faster also than the open space one. This last result can be influenced by the use of the percentage of free area covered as a termination condition. Moreover, in the comparison between the exploration of the obstacle environment and the open space, it has been found that the main varying parameter among the different strategies is the distance covered by the robot. It is much higher for the random strategy, and almost the same for the GB-L and the A-C-G, with a little advantage of the last one in the obstacle environment.

Finally, comparing the relation between the percentage of free area to cover and the distance travelled, it is shown that GB-L and A-C-G provide a linear growth, while random and greedy strategies don't.

Experiments runned show that greedy strategy has good performances only concerning the number of steps, not about the distance travelled. By taking into account also this metric, strategies considering also the cost of reaching a view-point perform better. Also the evaluation of the utility of a candidate observation position allows to improve the effectiveness of the exploration.

6 Frontier-based Exploration - Yamauchi

In this paper, a new approach for the exploration of unknown environments is introduced. It is based on the concept of frontier, which is defined as the region on the boundary between open and unknown space.

This new approach is defined over a robot using as spatial representation evidence grids and as sensor, a laser-limited sonar. It is a sonar whose measurements are integrated with the ones coming from a laser rangefinder and if the laser returns a range reading less than the sonar one, the evidence grid is updated with the laser reading, otherwise the sonar ones are used. This is done mainly to deal with flat surfaces or reflections and the bad readings they can produce for a sonar.

Once a frontier is detected, robot tries to navigate to the nearest accessible unvisited frontier and when it reaches its destination, performs a 360° rotation to update the evidence grid. If the robot doesn't succeed in reaching the frontier within a certain amount of time, that destination is marked as unaccessible and the robot computes the next frontier to reach.

Experiments were runned in two different environments: first one included an hallway and an adjacent office with furniture, second one consisted in large office area. In both cases, exploration ended quite quickly and in the first case, there are some frontiers which were marked as unaccessible because the robot took too much in reaching them. This is clarified by explaining that the first one is between a chair and a desk, while the second one is a narrow gap between two desks. Moreover, there are some frontiers which are not real, but produced by specular reflections mixed with the difficulty for the laser to detect them.

Advantages introduced with this frontier-based approach are three: the possibility of exploring environments containing both open and cluttered spaces; the possibility of exploring environments with walls and obstacles in arbitrary orientations and lastly, the efficiency with which exploration is pursued.

7 How much worth is coordination - Amigoni

Aim of this paper is to compare the contributions of different coordination methods and exploration strategies to an efficient exploration of an unknown indoor environment.

Two exploration strategies are used: one called *AOJRF strategy* and the other one called *MCDM strategy*. Both use 4 criteria to evaluate the utility function and they differ in how this utility function is defined. Criteria are: the amount of free area beyond the frontier; the probability that a robot will be able to transmit information to the base station, once reached the candidate location; the distance between such candidate location and the robot position; and the battery level of robot. The *AOJRF* defines an *ad hoc* utility function integrating the previously defined criteria. On the other hand, the *MCDM* one is way more complex and allows to consider criteria's importance and their mutual dependency relations.

Three coordination methods are used and they are called *AOJRF original coordination*, *AOJRF simplified coordination* and *no coordination*. It's important to point out that each method is runned by each robot indipentently, knowing the current map and the positions of the other robots. The first coordination method starts by computing the global utility of allocating each candidate location to each robot using Euclidean distance. After that, finds the pair *(candidate* *location, robot)* that maximizes such utility and recomputes the utility function only for this pair, substing the Euclidean measure of the distance with the one provided by the path planner. In the end, if this pair is still the best allocation, assigns that candidate location to the robot and removes them from the following computations as the loop restarts. The second coordination method is equal to the first one but for the re-computation of the distance through the path planner. In the third one, each robots only computes the best candidate location for itself, not taking into account the other robots.

Experiments were runned in an office and in an open environment, using two and three robots. Metric used to compare performances is the amount of safe area mapped in 15 minutes and also a random coordination is considered, expecting it to perform worse than others.

For the office environment, what comes out is that MCDM strategy always performs better than AOJRF, but differences are not statistically significant. Moreover, three robots always perform better than two, as expected. Analyzing more deeply coordination methods what can be seen is that AOJRF original coordination method and that of no coordination method are very similar and always better than that of the AOJRF simplified method, in both cases being statistically significant only for the MCDM. This is quite surprising because allocating tasks in a coordinated and in a not coordinated way produces similar results. It can be explained by pointing out that what matters in the exploration of an highly structured environment like an office, is the quality of the information used to evaluate candidate locations. In fact, the methods which works better are the ones using the distance computed by the path planner, not the Euclidean one; even if this quantity takes more time to be computed, it's totally worth it.

Also for the open environment, MCDM performs better than AOJRF. In this case, AOJRF simplified coordination method outperforms others in both exploration strategies with a statistically significance. This points out the importance of coordination in the case of an unstructured environment. Due to the fact that the number of candidate locations is lower with respect to the office case, coordination is needed to spread robots all over the map. On the contrary to the office experiment, the better performing method is the one using Euclidean distance, highlighting the smaller importance of the quality of information.