Decentralized Multi-Robot Exploration

yuntao.cai 1006519581

Supervisor Name: Goldie Nejat

April 14, 2022

Abstract

Mobile robots are rapidly becoming the ideal choice for exploring areas that are difficult or deemed too dangerous for humans to access. The task of mobile robot exploration can be modeled as the problem of selecting the next pose for exploration. A technique called map inference could potentially increase the efficiency of the pose selection problem as it uses known information provided by the robot to predict the surrounding unknown regions. The benefits of map inference are demonstrated in computer simulation as well as ROS simulation. In a structured office environment, the algorith showed an average 51% increase in available information, while having an accuracy of 78%. In a 3d, unstructured environment, the algorithm showed an average 48% increase in available information, while having a free and obstacle precision of 62%. This leads to a 36% decrease in total exploration time, as well as lead to a 45.37% decrease in total travel distance. The results of this algorithm have the potential to be utilized in search and rescue operations, resulting in a reduction of search time and facilitating the quicker identification of victims.

Contents

1 Introduction

Mobile robots are often used to explore and survey areas that are difficult for humans to access, such as mountainous terrains, caves, or disaster-stricken zones. Equipped with sensors such as lasers and cameras to gather data, mobile robots have become the ideal choice in many exploration tasks such as space exploration[1], search and rescue[2], and self-driving vehicles[3]. Therefore, the study of exploration algorithms has been a prevalent topic of robotic research.

The task of mobile robot exploration can be modeled as the problem of selecting the next pose for exploration. A good pose-selector would be able to choose the optimal position for the robot to explore next. This would make the robot increase coverage and to explore the unknown environment in a quick manner. This problem can then be further divided into sub-problems, such as selecting all poses of potential interest, determining the pose that is likely to gain the most reward, and in the case of multiple robots, how to settle disputes between conflicting objectives. Several algorithms exist for this pose-selection problem[5-10, 13-21], however there are many flaws such as inefficiencies in multi-robot coordination, poor estimate of pose rewards, and many others discussed in the Literature Review section.

The task of mobile robot exploration is further complicated when the need arises to deploy multiple robots. Multi-robot systems are desirable in certain situations as they can be built much smaller and lightweight compared to single robots, can explore more terrain in a given time frame, and can introduce an entire new set of strategies, such as sending a sacrificial robot to explore an area that is interesting but dangerous[4]. Currently, there are two main frameworks for controlling multiple robots, which can be classified as centralized[5] or decentralized[6][7][8]. These frameworks then can be further divided into classical or learning-based. The focus of this paper will be mainly on multi-robot, decentralized and classical approaches, however other methods are mentioned and critiqued in the Literature Review section.

One way to address to pose-selection problem is to use map inference, which is the approach that this report will take. Map inference algorithms uses the information that is obtained from robot sensors to reconstruct a cost map of the areas that it has already explored. It then uses this information to predict the terrain and structure of the surrounding unknown regions. Although there has been some prior research on map inference, this is still a largely untouched area of study. Previous work on map inference[9,10] has mainly focused on using visual features to recognize similar map segments between the current area under exploration and a library of fully explored map segments. This approach tend to be problematic because visual features of different map structures can be similar, leading to ambiguity in matching. Furthermore, these methods are computationally intensive and impractical in large-scale applications.

An efficient solution to the pose-selection problem has many significant implications, as it would allow multiple robots to search an unknown environment and identify objects of interest, for example victims in a disaster, with quicker time and less travel cost. Currently, the team that I am a part of is developing several algorithms for map-inference. We seek to replicate the results of several state-of-the-art algorithms as a benchmark[11-13], and also to develop a new algorithm that uses deep reinforcement learning to compare against the previous results. The goal of my research is to replicate an existing classical method for map-inference, and to compare this method against state-of-the-art algorithms. This report will include a literature review of the related background research, the methodology behind my choice of algorithm to implement, as well as results of the algorithm.

2 Literature review

This section will examine previous research in robot exploration and map inference, to show the limitations of current state-of-the-art papers, and select an algorithm for implementation.

2.1 Exploration algorithms

It was stated previously that robot exploration can be structured as solving the problem of which pose to select next for exploration. Having said that, not all robots use advanced algorithms for exploration, as demonstrated by commercially available robots like robotic vacuum cleaners which often employ the random walk algorithm[14]. This algorithm involves the robot moving in a straight line until it encounters an obstacle, then it would change its direction and the process is repeated. Evidently this algorithm is not very efficient and does not take full advantage of the data available. The concept of intelligent algorithms for exploration was first introduced by the Yamauchi paper[15], which proposed a frontier-based exploration algorithm. A frontier is defined as the boundary between known and unknown regions. A robot would first identify these frontier regions, and then it would be assigned the closest frontier point to explore. This process is repeated until the area has been fully explored or a new frontier becomes more accessible. This method is not efficient as it does not guarantee an optimal frontier to explore. In addition, this approach cannot be used in multi-robot exploration as it would lead to all robots converging onto a single goal. Market-based approaches, such as [16], have been deployed to coordinate the actions of the team with some levels of success. These approaches aim to optimize the balance between maximizing the rewards and minimizing costs, such as travel distance. They then select the pose with the largest overall benefits. However, while the cost of travel can be easily quantified, determining an exact metric for the rewards of exploring each pose is an ongoing area of research.

There have also been papers published on the subject of using the information gain to predict the reward of each frontier pose. Methods based on information-theory such as those outlined in references [17] and [18] have been shown to be effective in determining the potential information gain of each frontier location. The first paper mentioned above employs Rao-Blackwellized particle filters, where each particle represent a different trajectory that the robot could have taken. Each trajectory is then assigned an estimate of the potential information gain if the robot were to follow that path. Since unknown areas of the exploratory space has more entropy than known areas, the selected trajectory should reduce the entropy of the system to the lowest level, as this is considered equivalent to the most amount of information gain. The second paper utilizes mutual information to develop frameworks that approximate the information gain given by robots equipped with range-only sensors, as well as approximate representations of the costmap using the those sensor readings. Although both of these papers have good experimental results, their flaw is that they are only selecting the frontier points from areas that have already been observed. If the algorithm can predict the map structure of the unobserved environment with a high degree of certainty, then it can select frontier points in the unexplored regions of the environment. This would increase the efficiency of the robot as it is selecting frontier points with even more information gain. This technique, called map inference, is the main focus of this paper and is further discussed in detail in the Map Inference Section.

2.2 Multi robot exploration

Currently, there are two main frameworks for controlling multi-robot systems, which can be classified as centralized[5] or decentralized[6][7][8]. Centralized controllers are superior in terms of the speed of exploration and the total area covered than decentralized controllers, however it relies on perfect communication between robots themselves, as well as perfect communication between the robots and the central controllers. In practical applications, these controllers experience a significant decline in performance due to issues such as communication dropout or the failure of any individual robot[19]. In contrast, a decentralized system does not have an overarching network that oversees the entire system. Instead, each individual is acting on its own, relying on local communication of information. This approach may have poorer performance however it does not have a heavy computation load, and is also more resilient and robust to real-life considerations.

There are many control architectures for decentralized multi-robot exploration, ranging from classical approaches to learning-based approaches. Classical methods are are rule-based algorithms that uses hand-crafted heuristics that exploits certain properties of the environment. They have been the dominant method used in the past for their fast speeds and great performances in specific scenarios, however they can perform poorly in more generalized scenarios[20]. On the other hand, recent published papers have been increasingly utilizing learning-based methods, which are more versatile and can be applied to a wider variety of environments[21]. The main focus of this report will be on decentralized and classical approaches.

2.3 Map Inference

Map inference is used in robot exploration algorithms because if a robot can infer information about the unexplored portion of the map using the observed knowledge, then it can set up frontier points in the unexplored regions, which will increase the overall efficiency and speed of the exploration process.

Early contributions to the field include [9], which employs a Bayesian model to anticipate unexplored regions of a partially explored map, based on a library of maps that have already been fully explored. The algorithm, referred to as predictive-SLAM, selects a map from the library if it is similar enough to the current map, and the robot can use this predicted structure as a virtual mapping and determine frontier points. Another study [10] also employs a library of map structures to anticipate the current unexplored regions, but it attempts to identify loop-closures that the robot might encounter during exploration. This method uses a bag-of-words technique that identifies all possible matches that can be added to the map by comparing distinct visual features, then the best matches are merged onto the robot's current map using random sampling consensus. Although these two papers are successful at producing good experimental results, their methods of using visual descriptors of the environment is inefficient because vastly different map structures can have similar visual descriptors, which leads to ambiguity in matching. Additionally, these algorithms do not work in real-time, as comparing images is a computationally intensive task. To compensate, the algorithm would either compare enough images to produce an accurate result but run slowly, or it can compare fewer images and yield a less satisfactory outcome. Both options are not desirable.

2.4 Concept Selection

The team I am part of is currently working on a project that has three main papers of interest [11-13], the first paper being is a distributed map-inference algorithm, which is the main focus of this report. It is a classical algorithm that uses laser-based comparisons which are faster computationally and also less ambiguous than visual comparisons. The second paper employs neural networks that can anticipate the unknown regions of a partially explored map. It also incorporates some classical, information-theoretic techniques to improve the efficiency of the exploration process. The third paper uses a deep reinforcement reinforcement learning approach. The algorithm involves a centralized and a decentralized network during training phase, and only uses the decentralized network during testing phase. My thesis advisor, who is the author of this third paper, is working on an improved version of the deep reinforcement learning algorithm. The plan is to use the first two papers, which is the current state-of-the-art for classical and learning-based approaches respectively, as a bench-mark to compare the third approach against. Although it is expected the third method will be superior, there are still reasons why a classical method is a worthy comparison. Classical methods can work well during specific situations where the features of the environment is exploited and the heuristic is closely tailored to fit that environment. They are also faster to implement than learning-based algorithms. Lastly, learning-based algorithms tend to perform inadequately on map structures that they were not trained on. The following section explains the methodology of the classical, distributed map-inference algorithms, and then the results of preliminary trials are presented.

3 Methodology

3.1 Inferencing Algorithm

Before the distributed inference algorithm is described, some definitions must be introduced. The environment can be represented by a 2-D costmap, where each cell can have only one of 5 states. Observed-Free, Observed-Obstacle, Inferred-Free, Inferred-Obstacle, and Unknown. A cell is considered Observed when the robot uses its sensors(laser, camera, bump) that detected and identified the cell to be free or occupied. A cell is considered inferred when the algorithm determines the cell to be free or occupied not by observing it, but by predicting it using known information. All other cells are considered Unknown.

Firstly, a sparse 360 degree laser scan is performed at a desired location, p, on the costmap. How this location is exactly chosen is described in section 3.2. The laser scan is represented by an array of 360 numbers, where each number is the Euclidean distance between p and the first encounter of an obstacle cell, starting at 0 degree and incrementing by 1 degree. The laser scan is then compared to a library of fully-explored map structures, and the best matched library map-structure is determined using maximum likelihood. The library of map-structures was taken from [22] of the MIT campus.

For laser scan s and library scan l_i, the mean and standard deviation for both scans are calculated as μ_s, μ_l, σ_s, and σ_l respectively. The following approximate cumulative normal distribution function is introduced for it would be used later in the algorithm.

$$\Phi_s(z) \approx \frac{1}{1 + e^{-\sqrt{\pi}(-0.0004406x^5 + 0.0418198x^3 + 0.9x)}},$$

where

$$x = \frac{z - \mu_s}{\sigma_s}$$

Then, for a laser scan s and a library scan l_i, we have two arrays to compare against each other. Denote each individual beam of the array as s_j and l_{ij} respectivey, the probability that a random beam in s being generated by library beam l_i is given by

$$P(s_j = l_{ij}) = 1 - |\Phi_s(\lambda_1) - \Phi_s(\lambda_2)|$$

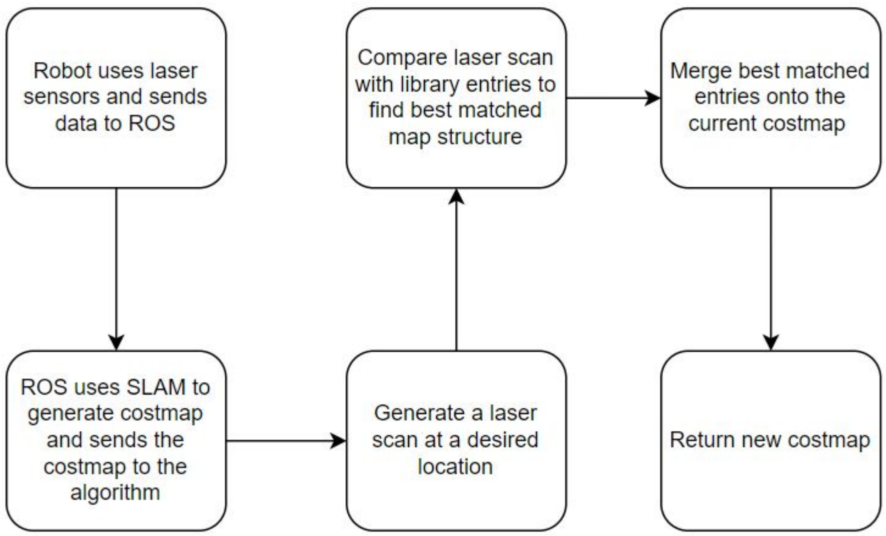
where

$$\lambda_1 = s_i + \gamma, \lambda_2 = s_i - \gamma, \gamma = |l_{ij} - s_i|$$

Note that as γ approaches 0, $P(s_j = l_{ij})$ approaches 1. Using this, the total probability that the map structure l_i has generated s is given by

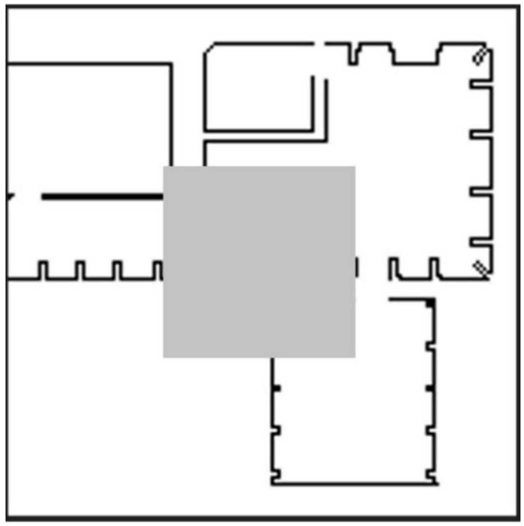
$$\sum_{j=1}^{|s|} -log(P(s_j = l_{ij}))$$

Logarithm is used for faster computation. Library entries that have exceed a certain user defined threshold will then be merged onto the current environment. Here is a control diagram describing the general architecture of the algorithm.

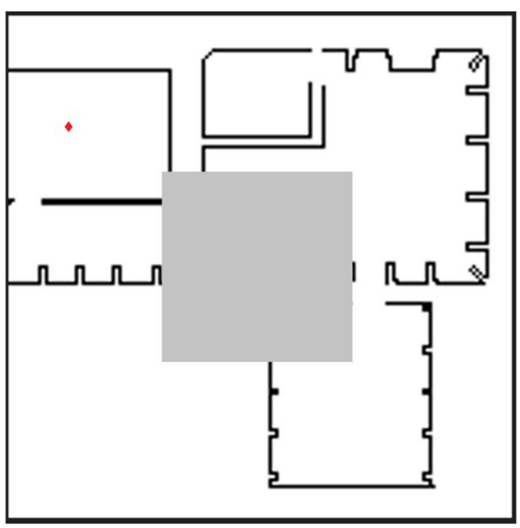


A more detailed example is given below.

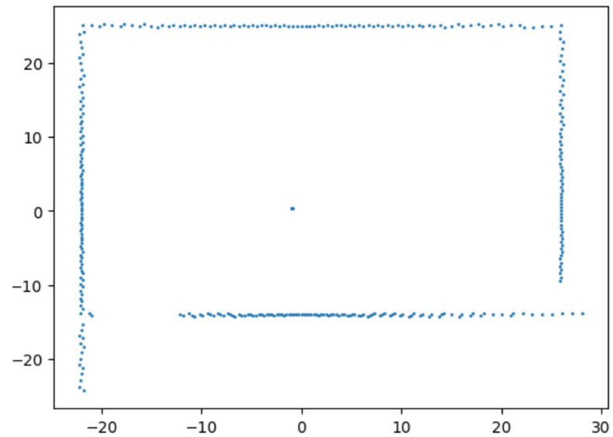
In this scenario, a partial map is presented, and the objective is to predict the structure of the grayed out areas in the map.



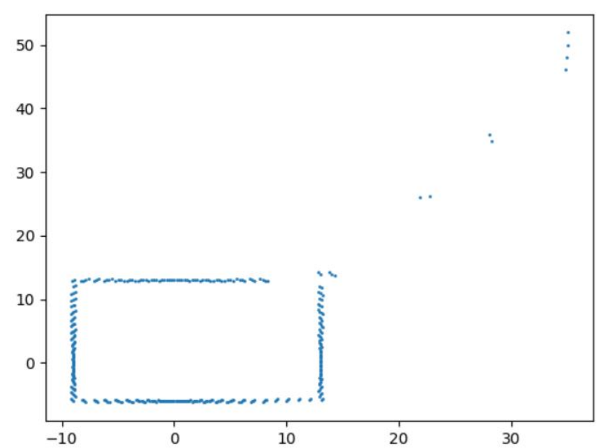
A desired location on the partial map is chosen for performing a laser scan, which is marked by a red rhombus.



A sparse 360 degree laser scan is performed. Notice the similarity between the room where the red rhombus is located and the laser scan.



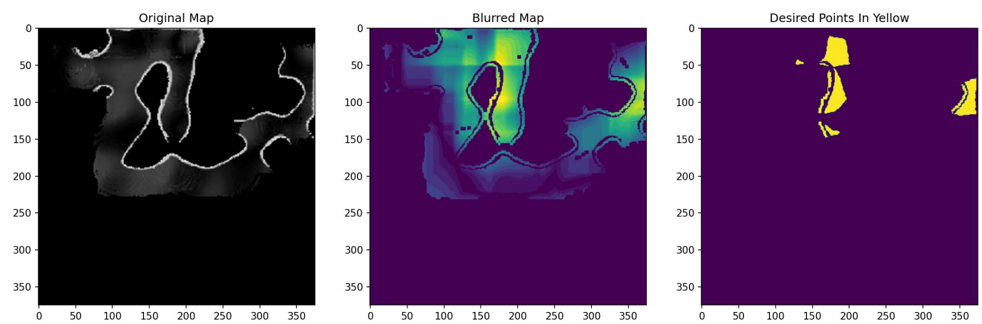
Now a search through a library of previously recorded map structures was conducted. It was found, through the maximum likelihood evaluation criteria above, that the laser scan below has the closest match to the laser scan above



The map structure is then merged onto the existing environment. The results of which can be seen in section 4.1.

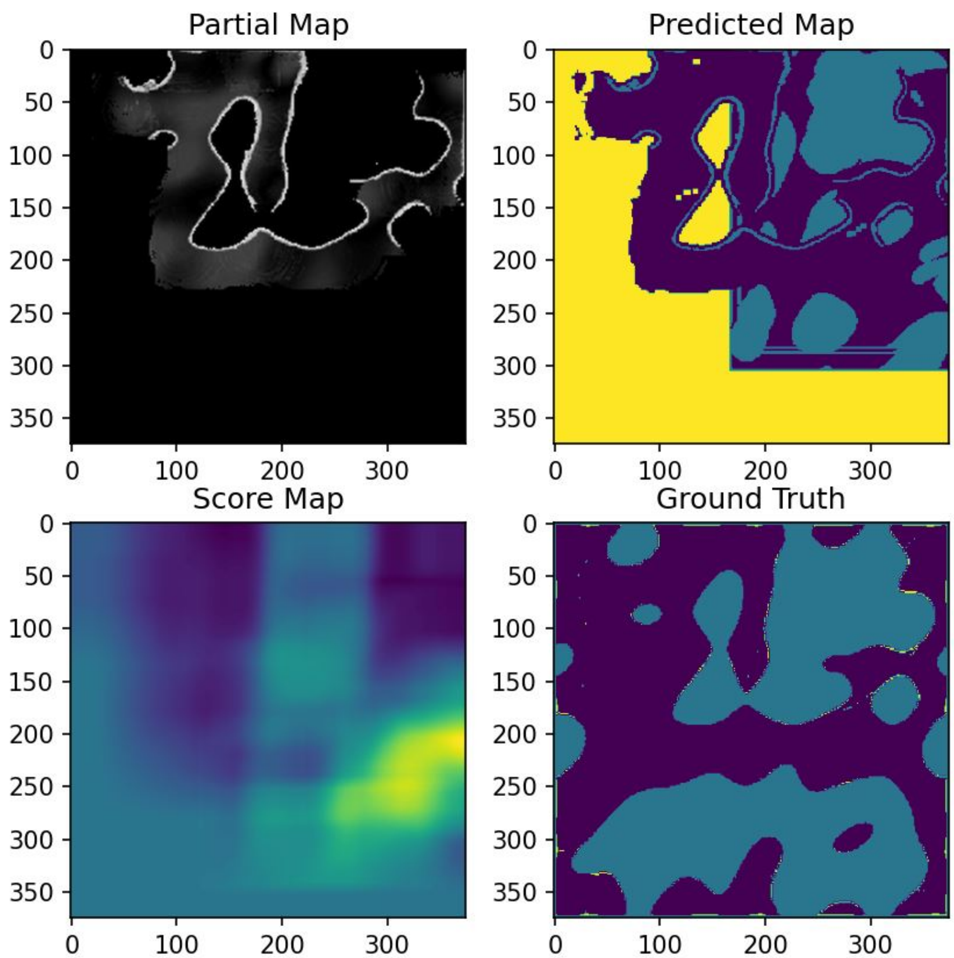
3.2 Choosing Points for Laser Scan

This section explains the methodology behind how to choose a point for laser scan. It was found through trial and error that the desired point should have surrounding obstacles, since an open area or an unknown area would return no laser scan. In order to achieve this, the map is first blurred. The blurring process combines the free and obstacle cells in the map by smoothing them out, resulting in a more averaged representation. Then, the average value of the map is calculated, and the difference between the average value and the value of each cell is taken. The conjecture is that the closer the value is to the mean, the more likely it is to be between obstacle cells. The cells below one standard deviation are chosen(as shown in yellow), and a point is randomly selected to be used for laser scanning.



3.3 Choosing Frontier Points

Once the predicted map has been obtained, it remains to select frontier points on the map used for exploration. The choosing of frontier points would use the available occupancy grid of known cells, as well as the newly predicted costmap to make a selection. A score map is created using these information, such that each individual cell is given a score, prioritizing Inferred free cells and Unknown cells; Observed free cells have lower score, and Inferred Obstacle and Observed Obstacle cells have no score. Blurring is done on the score map for averaging purposes, and the top scoring cells are more likely to be selected for exploration.



For the example above, based on the partial map information, the predicted map is shown on the top right. The blue colour represents Known and Inferred Obstacles, the purple represents Known and Inferred Free cells, and the yellow represents the Unknown. The score map is shown on the bottom left, where a brighter colour means a higher likelihood to being explored. Notice that in this case, the brightest area, which is the area that is most likely to being explored, was not observed by the robot prior to inference.

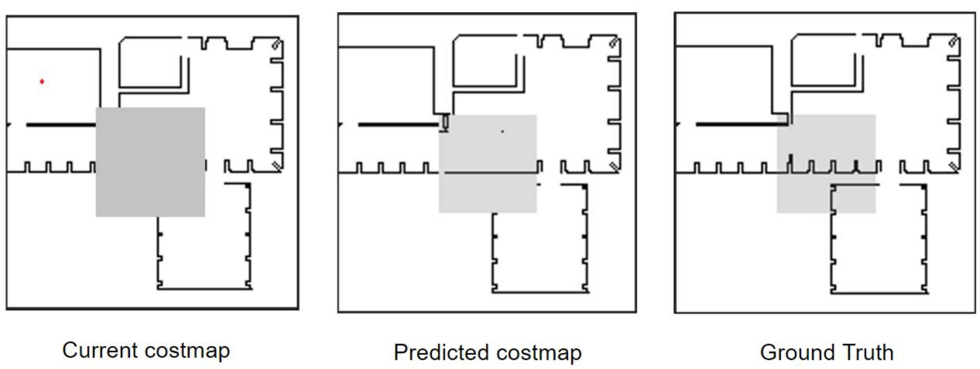
Selecting a point beyond the observable area offers higher potential reward by reaching that location.

4 Results

To assess the performance of the algorithm, two sets of computer simulation as well as two ROS simulation was conducted. The first computer simulation and ROS simulation was done the Aydemir data set[22], which is a library of 2D office building floor plans. The second computer simulation as well as the ROS simulation was conducted on a custom, 3D unstructured environment.

4.1 Computer and ROS Simulation - 2D

The 2-D simulations were mainly done as a proof of concept, to see if we can obtain the same results as the authors of the algorithm had[11]. Partially explored maps were taken and the structures of the unknown areas was predicted upon. The results below was obtained from conducting experiments on 200 different partial map environments.

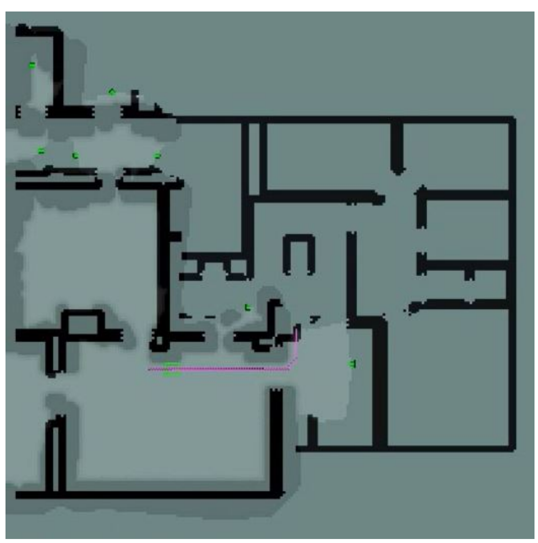


Accuracy	78%
Generous accuracy	83%
Obstacle precision	10%
Obstacle recall	13%
Obstacle generous recall	14%
Free precision	92%
Free recall	84%
Free generous recall	89%

Accuracy represents the number of correctly predicted cells, both Free and Occupied. Obstacle precision refers to the proportion of the predicted obstacle cells that were actually true obstacle cells, and free precision is the same but for free cells. Recall means how much new information was gained. Any evaluation criteria with the word generous means it was computed in proportion to the known cells, instead of the total number of cells. Since the number of known cells is always less than the total number of cells, this value will always be greater than its non-generous counterpart, hence why it is referred to as generous.

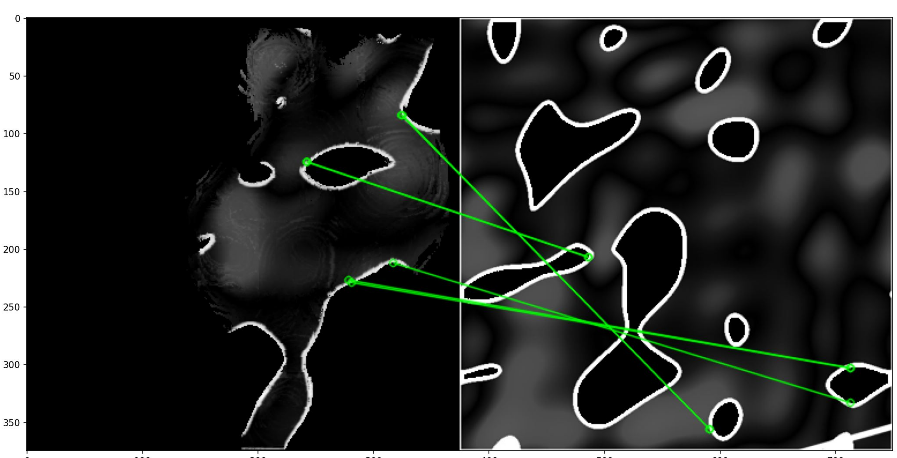
The algorithm on average has an accuracy of 78%, with an obstacle cell precision of 10%, obstacle cell recall of 13%, free cell precision of 92%, and free cell recall of 84%. Free cell precision and recall is high due to the nature of the environment, which consists of mainly office buildings. Since there are more free space as opposed to obstacle space, the metrics are biased towards free cell accuracy but lacking in obstacle cell accuracy. It should also be mentioned that the free precision is very similar to the results of 95.1% that was obtained in the original paper[11].

Since the initial proof of concept proved promising, we moved onto a simulator in ROS. While the robot is traversing an unknown environment, the predictions made by the robot is shown below. The black lines indicate the map structure inferred by the algorithm. The simulation proved that the map-inference algorithm decreased exploration time by 10%.



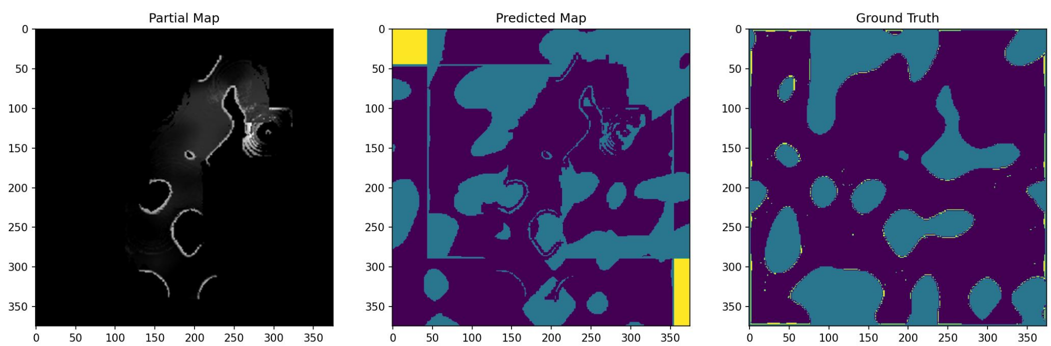
4.2 Computer Simulation - 3D

Since the initial 2D tests showed promising results, the project has been extended to 3D, unstructured environments. Unlike 2D environments, where obstacles and free space can be represented in binary, 3D environments are represented by images. The x and y coordinates are represented by the array of the image, and the pixel intensity of the image represents the height. The task of finding the best matched map structure can be then formulated as finding the image matches the current environment most accurately. Therefore, this is very similar to an image-recognition and object detection problem. However, typical image matching techniques such as SIFT and ORB cannot be easily applied in this scenario, since they both use local descriptors for object identification. One problem that often arises with the use of local descriptors is that the SIFT or ORB algorithm will often find a local feature that matches, however it does not match on a bigger scale (See the image below). Therefore, this method proves invalid in the inference process in that it could not find a map structure that is similarly enough to the current environment.



In such a situation, the algorithm above could actually be used to find a good match for map inference. The height map can be approximated by taking all heights that are traversable as "free cells", and all heights too tall for the robot as "obstacle" cells. A laser scan can then be emitted, and a similar map structure could be found. Inadvertently, the above algorithm actually provides an alternative image-recognition technique, however its scope is limited in that it can only work in situations where the image in question can be converted into binary form.

The 3D computer simulation uses the same technique as the 2D computer simulation. A laser scan is performed on a partial map, and that scan is compared against a library of laser scans. Maximum likelihood is similarly used to determine the best map structure, which will then be merged onto the current environment. 120 trials were conducted, and the results of the 3D computer simulation are given below.



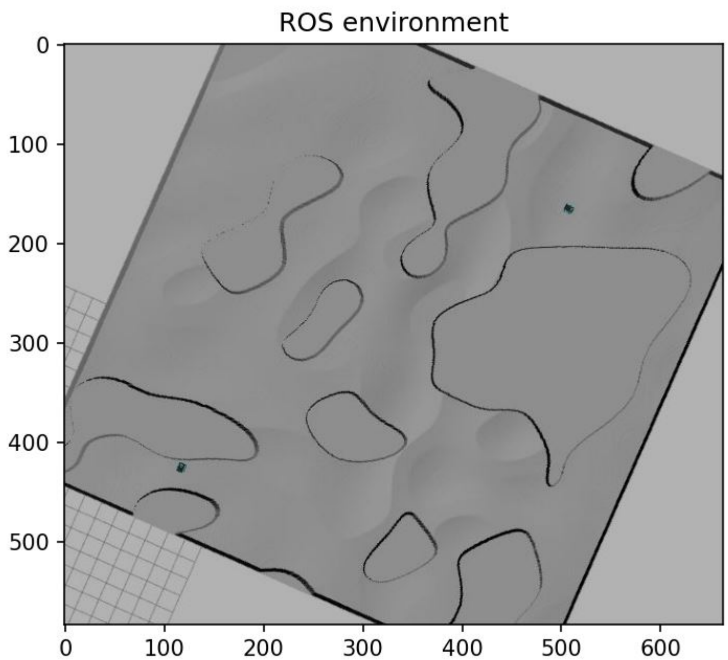
Accuracy	41%
Generous accuracy	54%
Obstacle precision	63%
Obstacle recall	57%
Obstacle generous recall	63%
Free precision	61%
Free recall	39%
Free generous recall	70%

The results of these trials are different from the 2D computer simulation mainly due to changes of the environment. The 3D, unstructured environment has roughly an equal amount of free and obstacle cells, where as the 2D environment is mainly free cells. This bias toward free cells is what caused the precision of free cells to be so high, and why the precision of obstacle cells is so low. The accuracy of the 2D computer simulation is therefore higher mainly due to the high precision of the prediction of free cells. However, in the 3D environment there is no obvious bias towards either free or obstacle cell, that is why the precision for obstacle and free cells are similar.

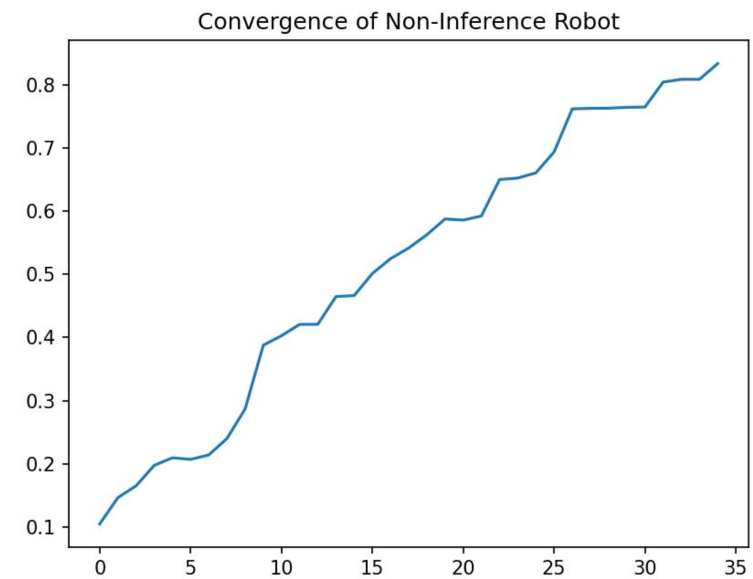
Although it may seem that the algorithm does not perform quite well for the 3D unstructured environments, a large part of that is due to the flaws of the evaluation criteria. Accuracy, recall and precision are standard ways of measuring how well a map-inference algorithm performs, and it is also what the original authors had used. Nevertheless, the evaluation criteria is binary in nature, in that the algorithm will either predict a cell as free or obstacle, and nothing in between. Yet in reality, the algorithm returns not a binary value but rather a probability distribution of how likely the cell is to be free or occupied. This value is then separated using a threshold to be either free or obstacle. A better evaluation function would be a probabilistic based function which takes into account that each cell has a non-binary value.

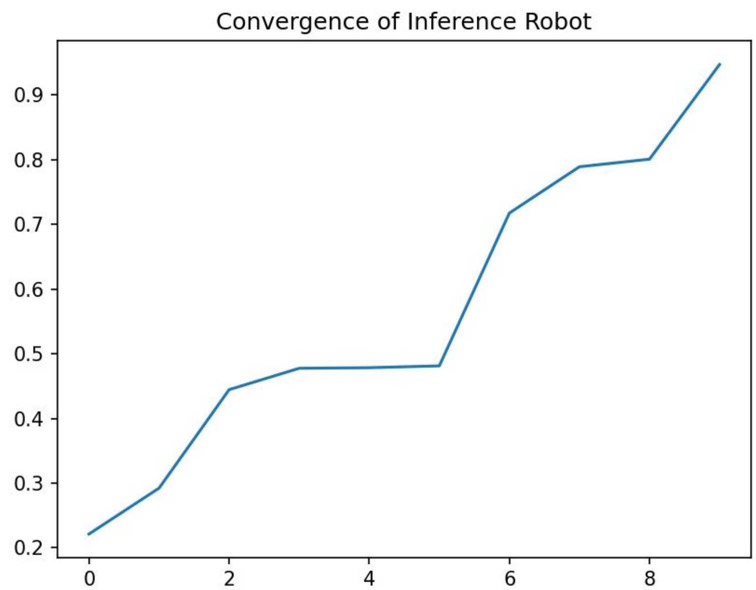
It is also desirable, for exploration purposes, to have a high precision in both free and obstacle cells. It is not desirable to have one being high and the other being low. A high precision in free cell but a low precision in obstacle cell would result in situation such as having unplanned obstacles blocking the way of the robot. A low precision in free cell but a high precision in obstacle cell could result in the robot being stuck in one place because it predicts it is surrounded by many obstacles. For the results of the 3d unstructured environment, although the individual precision is lower compared to structured environments, they are very similar to each other, and are larger than 50%. This means that a portion of the unknown environment is being predicted correctly that the exploration process can take advantage of while selecting frontiers. As demonstrated in the ROS simulation results, even with not so high precision, the algorithm can perform better than naive methods for frontier exploration.

4.3 ROS Simulation - 3D



ROS simulation was also conducted on the 3D costmap. The performance of the map-inference exploration method was compared with that of the naive frontier exploration method to demonstrate its superiority. It was found that the map inference method decreased exploration time by 36%, as well as lead to a 45.37% decrease in total travel distance.

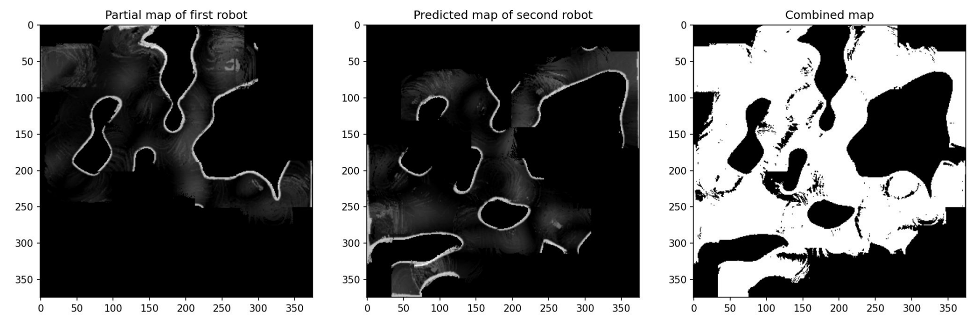




The exploration algorithm converges if the robots combined have explored 85% of the environment. The convergent rate for a non-inferencing robot to completely explore an environment is typically linear, while the convergent rate of the inference robot has a sharp increase in the middle due to inferring a frontier point with a large reward. The inference robot image below shows a trial run where the robot had predicted a frontier point on the other side of the map, therefore resulting in a significant reward gain, and the exploration algorithm converged when the robot only had to explore 9 distinct positions. In contrast, the non inferencing rate of convergent is much slower, as it iteratively finds frontier points only near its vicinity and does not infer the unknown area. Hence why it shows a more gradual convergence.

Although this result is better than the performance of the original paper of 13.15% decrease in total travel distance, several things must be taken into consideration:

First of all, the inferenced area was very large compared to the total area of the environment. This means that in some situations, a robot would find an inferenced frontier point with a very large reward. The original authors had much larger map sizes compared with our own, and the inferenced area is much smaller. Therefore, since the robot only infers a smaller area, the reward of its frontiers are smaller. Additionally, the ROS simulation was done on a computer, while the original authors of the algorithm had conducted hardware trials. Hardware trials are much more prone to error, as real life consideration such as sensor noise, speed of computation of algorithm, and conflicting robots could potentially increase the exploration distance.



5 Conclusion

The main algorithm implemented in this paper is the map inference algorithm defined by [11]. It has been shown through computer simulation that it is possible to predict the unknown areas of the environment with the known information. Additionally, the efficacy of the proposed map inference method has been demonstrated through ROS simulation. The result is a 36% decrease in exploration time, as well as a 45.37% decrease in total travel distance.

Several drawbacks of the algorithm should be noted. One of the fundamental limitations of using a library of map structures is that it is difficult to generalize to a completely new environment. Although the original paper[11] claims that any environment can be approximated by the library of map structures, the results of this paper shows that it cannot be done as effectively. In addition, the algorithm requires preexisting map structures to be loaded for it to run, which could increase the overhead to the runtime of the exploration process.

The implementation of this algorithm is expected to be bench marked against algorithms which the team I am a part of will also create. Although it is expected that this algorithm will not reach the efficiency of machine learning algorithms, there are still advantages to this algorithm. It is can be implemented much quicker than any machine learning algorithms, and is comparable in run time to machine learning algorithms. Machine learning algorithms also tend to run into over-fitting problems, which is non-existence in this classical, map inference algorithm. In niche areas that do not have the hardware necessary for machine learning, this algorithm offers an alternative solution to map inference and robot exploration

References

[1] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson, and T. Smith, “Autonomous Rovers for mars exploration,” 1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403), 1999.

[2] Y. Liu and G. Nejat, “Multirobot Cooperative Learning for semiautonomous control in Urban Search and Rescue Applications,” Journal of Field Robotics, vol. 33, no. 4, pp. 512–536, 2015.

[3] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 33–55, 2016.

[4] R. A. Brooks, Flesh and machines: Robots and people. New York: Pantheon Books, 2002.

[5] F. Gul et al., ”A Centralized Strategy for Multi-Agent Exploration,” in IEEE Access, vol. 10, pp. 126871-126884, 2022, doi: 10.1109/ACCESS.2022.3218653.

[6] Z. Yan, A. A. Cherif, and N. Jouandeau, “Multi-robot decentralized exploration using a trade-based approach,” Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics, 2011.

[7] A. Batinović, J. Oršulić, T. Petrović, and S. Bogdan, “Decentralized strategy for cooperative multi-robot exploration and mapping,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 9682–9687, 2020.

[8] J. P. Queralta et al., “Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision,” IEEE Access, vol. 8, pp. 191617–191643, 2020

[9] H. J. Chang, C. S. Lee, Y.-H. Lu, and Y. C. Hu, "P-SLAM: Simultaneous localization and mapping with environmental-structure prediction," IEEE Transactions on Robotics, vol. 23, no. 2, pp. 281–293, 2007

[10] D. P. Strom, F. Nenci, and C. Stachniss, "Predictive exploration considering previously mapped environments," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015.

[11] A. J. Smith and G. A. Hollinger, "Distributed inference-based multi-robot exploration," Autonomous Robots, vol. 42, no. 8, pp. 1651–1668, 2018.

[12] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for Enhanced Mobile Robot Exploration," 2019 International Conference on Robotics and Automation (ICRA), 2019.

[13] A. H. Tan, F. P. Bejarano, Y. Zhu, R. Ren, and G. Nejat, "Deep reinforcement learning for decentralized multi-robot exploration with macro actions," IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 272–279, 2023.

[14] A. EREN and H. DOĞAN, "Design and implementation of a cost effective vacuum cleaner robot," Turkish Journal of Engineering, 2021.

[15] B. Yamauchi, "A frontier-based approach for autonomous exploration," Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', 1997.

[16] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Feb. 2002.

[17] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using Raoblackwellized particle filters,” Robotics: Science and Systems I, 2005

[18] B. Charrow, V. Kumar, N. Michael, C. J. Taylor, D. D. Lee, A. Ribeiro, and M. Schwager, “Information-theoretic active perception for multi-robot teams,” dissertation, 2015.

[19] .N. Seenu, R. M. Kuppan Chetty, M. M. Ramya, and M. N. Janard-hanan, “Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems,” Ind. Rob., vol. 47, no. 6, pp. 929–942, 2020.

[20] M. Geng, K. Xu, X. Zhou, B. Ding, H. Wang, and L. Zhang, “Learning to cooperate via an attention-based communication neural network in decentralized multi-robot exploration,” Entropy, vol. 21, no. 3, 2019

[21] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, A survey and critique of multiagent deep reinforcement learning, vol. 33, no. 6. Springer US, 2019.

[22] A. Aydemir, P. Jensfelt and J. Folkesson, "What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 2012, pp. 4675-4682, doi: 10.1109/IROS.2012.6386110.