

Multi-Robot Informative Path Planning in Unknown Environments through Continuous Region Partitioning

Ayan Dutta,¹ Amitabh Bhattacharya,¹ O Patrick Kreidl,¹ Anirban Ghosh,¹ Prithviraj Dasgupta²

¹University of North Florida, USA; ²University of Nebraska at Omaha, USA

{a.dutta, a.bhattacharya, patrick.kreidl, anirban.ghosh}@unf.edu, pdasgupta@unomaha.edu

Abstract

Information collection is an important application of multi-robot systems especially in environments that are difficult to operate for humans. The objective of the robots is to maximize information collection from the environment while remaining in their path-length budgets. In this paper, we propose a novel multi-robot information collection algorithm that uses a continuous region partitioning approach to efficiently divide an unknown environment among the robots based on the discovered obstacles in the area, for better load-balancing. Our algorithm gracefully handles situations when some of the robots cannot communicate with other robots due to limited communication ranges.

Introduction

Dynamic path planning is a well-known and crucial aspect of autonomous navigation of multiple mobile robots. Recently, researchers have considered a practical aspect of the multi-robot path planning problem where each robot has to dynamically plan a path which helps them to collect maximal possible information from the environment, known as the *Multi-robot Informative Path Planning* (MIPP) problem (Singh et al. 2007). The collected information can be of various types - from simple images captured by an on-board camera to radiation measurement in a nuclear power-plant. In real-life, robots have limited battery power and might intermittently go out of communication ranges of each other. Thus, finding an optimal solution for the above-mentioned problem becomes notoriously difficult – an NP-hard problem (Hollinger and Singh 2010). We consider an additional navigation criterion for information collection – no two robots should visit the same region of the environment if they already know about the others’ visited regions.

To address these challenges, we have proposed a dynamic region partitioning approach where the robots with limited communication ranges can efficiently plan distributed informative paths and when required, repartition the allocated regions among themselves for better load balancing of the information collection task. Moreover, we consider the MIPP problem in unknown environments where the robots do not

have any knowledge about the shapes and locations of the obstacles. Our proposed approach solves this notoriously difficult problem in such a way that although the robots might go out of each other’s communication ranges intermittently, whenever they are able to communicate with each other, they plan to recursively partition the region based on their current perceptions about the environment so that redundant information collection can be avoided and a better load-balancing of the task can be achieved.

In our knowledge, our proposed approach in this paper is one of the firsts to solve the MIPP problem under communication constraints in unknown environments using a Voronoi partitioning-based approach that aims to divide the region among the robots for better load-balancing and minimized redundant information collection. An illustrative example scenario is shown in Figure 1.

Related Work

One of the very first studies on the MIPP problem is due to (Singh et al. 2007). This work uses Gaussian Processes (Rasmussen and Williams 2006) to model the environmental spatial phenomena. They have proposed a branch-and-bound based budgeted path planning technique to solve the problem. The MIPP problem with the robots having periodic connectivity has been studied in (Hollinger and Singh 2010). In (Cao, Low, and Dolan 2013), the authors have proposed two greedy strategies for information collection. Our work in this paper uses a similar greedy path planning strategy for information collection. Sampling-based path planners have also been used for information collection (Hollinger and Sukhatme 2014) where the authors have proposed a variant of the RRT* algorithm (Karaman et al. 2011).

Voronoi partitioning is a scheme of decomposing an area into disjoint cells using the “nearness” concept (Voronoi 1907). Voronoi partition has been used in solving several problems in robotics. Choset and Burdick (Choset and Burdick 1995) use a generalized Voronoi diagram for a robot path planning problem. In (Hungerford, Dasgupta, and Guruprasad 2016), the region to be covered by a group of robots is decomposed into Voronoi cells and each robot is responsible for covering the corresponding Voronoi cell. Moreover, if part of one robot’s allocated cell is inaccessible to that

robot, other robot(s) covers the inaccessible part of the cell. Similar area coverage strategy using a group of networked robots has been studied in (Breitenmoser et al. 2010).

This paper aims to bring the above-mentioned concepts together: use the idea of region partitioning for better load-balancing among the robots for information collection. The closest study to this work has recently been published in (Kemna et al. 2017). Our work is complementary and extends the direction of the work in (Kemna et al. 2017) in the following way: the work in (Kemna et al. 2017) does not consider any obstacle in the environment whereas our repartitioning algorithm works based on robots' explored regions and the unknown obstacles discovered by them.

Model

Let \mathcal{R} denote a set of n robots $\{r_1, \dots, r_n\}$. We assume that the robots know the boundaries of the environment but have no initial knowledge about the location or geometry of obstacles in the environment. To detect obstacles during operation, each robot is equipped with a laser rangefinder with a given detection range. The robots are localized in the environment (e.g., using GPS). Each robot can communicate its location, path and obstacle information with other robots only if within a given radio range. Each robot is also equipped with an information collecting sensor e.g., camera. We assume robots whose radio range (for communication) is greater than the detection range (for obstacles), the latter also greater than the sensing range (for information).

The robots operate within a convex, polygonal environment denoted by $E \subset \mathbb{R}^2$, consisting of free and inaccessible (due to static obstacles) sub-regions, $E = E_{free} \cup E_{obst}$. The obstacles are convex polygons as well. A set of information collection points cover E in a grid structure. This set of locations are called Points of Interest (POI). The robots move from one $poi \in POI$ to the other in order to collect information. Once the robot reaches a point of interest, it uses its sensor(s) to collect necessary information from the surroundings of their current poi . The POI set can be decomposed into two disjoint subsets, V and U , corresponding to the locations that are visited and not visited by a robot. Each robot is assigned a budget B , which indicates the maximum number of poi 's each robot is allowed to visit. The objective of every robot is to follow a maximally informative path, minimizing redundancy in information collection subject to containing its path length within the given budget.

Methodology

Voronoi Partition Voronoi partition is a widely used mechanism for partitioning a space based on the "nearness" concept. Given a convex polygonal region E and a set S of n sites $\{s_1, \dots, s_n\}$, we can associate a polygonal Voronoi cell V_i with every site $s_i \in S$ as the following:

$$V_i = \{q \in E : \|q - s_i\| \leq \|q - s_j\|, \forall s_j \neq s_i \in S\} \quad (1)$$

where $\|q - s_i\|$ denotes the Euclidean distance between two points $q, s_i \in E$. Intuitively, each Voronoi cell V_i is the collection of those points which are closer to s_i than any other site $s_j \in S$. This Voronoi cell is assigned to a unique r_i for

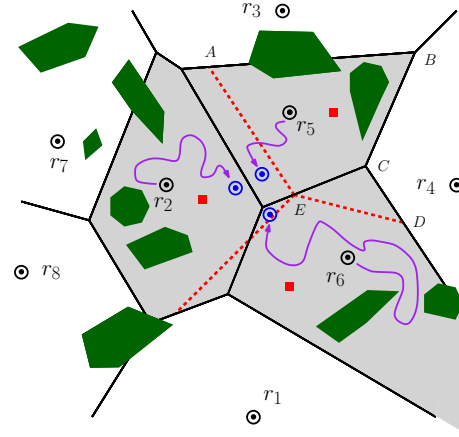


Figure 1: Eight robots (circles) navigate through their initial Voronoi partitions (separated by black lines) of an environment with obstacles (green polygons). Robots r_2 , r_5 and r_6 eventually move within communication range (blue circles), where the union of their respective cells (shaded gray) is repartitioned (separated by red-dashed lines) to rebalance remaining collection load based upon their updated perceptions on the free versus obstructed subregions.

information collection; r_i will only collect information from V_i . The intersection of any two Voronoi cells is either null, a line segment, or a point. Each Voronoi cell is a topologically connected non-null set. This standard partitioning of the environment into n Voronoi cells for n sites can be done in $\mathcal{O}(n \log n)$ time (Steven 1987).

Initial Partitioning We assume that the robots are dropped from an aircraft and based on their dropping locations, an initial Voronoi partitioning of the environment is done and the respective map is stored in every robot. In this case, S is the set of initial locations of the robots. However, the robots might not be immediately able to communicate with each other after being dropped because of the distances among them and also due to lack of knowledge about the obstacles in the environment at this point.

Informative Path Planning Once the initial region partitioning is done and all the robots are allocated to specific regions to collect information from, they will plan informative paths within their own Voronoi cells.

Gaussian process (GP). To model the environmental phenomena generating the information and corresponding informative paths for the robots, we have used Gaussian Processes (Guestrin, Krause, and Singh 2005). GP assumes that all the information collection locations (POI) in the environment have a joint Gaussian distribution. A GP is defined by its posterior mean $m(\cdot)$ and variance $\sigma^2(\cdot)$ functions. Given a set of measurements X_V , sensed from the visited poi 's, we can predict the information measurement in the rest of the unobserved locations U , conditioned on X_V . A GP can be specified by the following equations (Guestrin, Krause, and Singh 2005):

$$\begin{aligned} \mu_{U|X_V} &= \mu_U + \Sigma_{UV} \Sigma_{VV}^{-1} (X_V - \mu_V) \\ \sigma_{U|X_V}^2 &= \Sigma_{UU} - \Sigma_{UV} \Sigma_{VV}^{-1} \Sigma_{UV} \end{aligned} \quad (2)$$

where $\mu_{U|X_V}$ is the conditional mean and $\sigma_{U|X_V}^2$ is the variance. Σ_{UV} is the co-variance matrix, which contains an entry for every visited $poi \in V$. Following the GP formulation, the objective of informative path planning is to plan a path which maximizes the posterior entropy where entropy can be formally defined as follows:

$$H(u|X_V) = \begin{cases} \frac{1}{2} \log(2\pi e \sigma_{u|X_V}^2), & u \in E_{free} \\ -PEN, & \text{otherwise} \end{cases} \quad (3)$$

The objective of entropy maximization is to find the unvisited locations in the environment which contain the highest amount of uncertainty, i.e., predicting the information measurement in these locations are highly uncertain and inaccurate. However, if the unvisited location is not obstacle-free, we assign a negative penalty value as the entropy value for u to ensure that no robot ends up at u .

Path Planning. Each robot's objective is to reduce the uncertainty in the environment. The robots need to visit the locations with the highest uncertainty and consequently the highest entropy. To do this, each robot follows a greedy informative path generation strategy that calculates the next best poi (i.e., the highest entropy location) for it to visit within its own allocated region based on the previously visited- poi set V . The greedy path planning strategy has been proved to yield good results in spatial modeling (Cao, Low, and Dolan 2013; Kemna et al. 2017). Formally, we can write that each robot r_i will select an informative location poi_{next} within its own region E_i to visit next using the following formula:

$$poi_{next} = \arg \max_{poi \in E_i} H(poi|X_V) \quad (4)$$

Note that, poi_{next} is a neighbor- poi of the robot's current informative location. Given the fact that each robot calculates only the next best poi to visit in every iteration (i.e., the planning horizon is 1) in a distributed manner (i.e., calculates only its own path) and poi_{next} is a neighbor poi of the robot's current location, the above-mentioned path planning strategy will scale polynomially on the size of V (Cao, Low, and Dolan 2013). The robots move in a straight line between two poi 's in a deterministic manner.

Recursive Region Partitioning Each robot, r_i , will continue to generate a new path within its designated Voronoi cell and will follow that path until one of the following two things happen: 1) The path length covered so far by r_i exceeds the assigned budget, B , or, 2) r_i meets (i.e., comes within the communication range of) another robot r_j .

While visiting new poi 's within its designated region, robot r_i will continuously broadcast messages containing its unique ID, current location and current allocated region, i.e., V_i , a polygonal Voronoi cell. If r_i meets another robot(s), then depending on received information from the other robot(s), they will decide whether to repartition the union of their current Voronoi cells to achieve load-balancing or to stay within their currently designated cells.

To keep track of contacted robots and their corresponding Voronoi cells along with their discovered obstacles, robot r_i maintains a data-structure named Perception of World (PoW_i). The PoW_i can be imagined as a 2D array of time-stamped objects indexed by poi 's and contains information about the following: V_i 's, discovered obstacle locations,

Algorithm 1: Recursive region partitioning for information collection

```

1 Each robot  $r_i$  executes the following procedures.
2 Procedure pathPlanning()
3   if Covered budget-limit distance  $B$  then
4     Broadcast the latest  $PoW_i$  along with a FINAL
       message to indicate that the budget-limit is
       reached.
5   else
6     Plan the next  $poi$  to visit in its own partition using
       Equation 4.
7     Move to this  $poi$  location to collect information.
8 Procedure RecursivePartitioning()
9   Create a data-structure,  $PoW_i$ , to keep track of the
       current perception about all the  $poi$ 's.
10  if meets a robot  $r_j$  with which  $r_i$  has never met OR
        $PoW_i$  and/or  $PoW_j$  have changed from the last time
       these robots met then
11    Update the corresponding Voronoi partitions.
12  Plan the next  $poi$  to visit for information:
       pathPlanning().

```

sensed information measurement and the variance in prediction. It might happen that robot r_i discovered obstacles in robot r_j 's current Voronoi cell, which r_j might or might not have any knowledge about. But when they meet, both of them will have the knowledge about these obstacles.

As the obstacles are assumed to be static, if PoW 's are to be exchanged between r_i and r_j , this perception will be shared by them and corresponding repartitioning strategy will be implemented. Note that, PoW_i is a local data-structure, i.e., different robots' PoW_i 's might be different in terms of stored content depending on which robot(s) they came into contact with. This data-structure will help the robots to partition their respective regions if required in a more intelligent way as we will see next.

Let \bar{R}_i denote the set of robots which are within robot r_i 's communication range. If r_i meets with $r_j \in \bar{R}_i$, one of the following situations can arise:

1. r_i has never communicated with r_j before.
2. r_i has communicated with r_j communicated before.
 - (a) r_i and/or r_j 's perceptions about the other robots' partitions (PoW_i and/or PoW_j) have changed after r_i last communicated with r_j .
 - (b) PoW_i and PoW_j are still the same from r_i and r_j 's last communication.

We will handle these cases one by one. We will start off with the simplest case (2.b) – r_i and r_j 's perceptions about the other robots' partitions (PoW_i , PoW_j) have not changed from the last time these two robots have met. In this case, they don't have to update their plans and/or their respective Voronoi cells and they can plan their next best poi 's to visit within their current Voronoi cells (line 12 in Algorithm 1).

If r_i and r_j are meeting each other for the first time (case 1), they might or might not have current partition information about each other. In any case, they will repartition their

current Voronoi cells based on their latest knowledge about the obstacles in their respective *PoW*'s. As the environment is unknown to the robots in the beginning and they are discovering shapes and locations of the obstacles in the environment as they explore more, the initial Voronoi cells might not be 'balanced' in terms of the free space available to explore. Consequently, one robot might be tasked with exploring significantly larger/smaller amount of region in the environment compared to the others. In order to minimize this potential high level of variance and bring balance to the obstacle-free areas of robots' exploration regions, we use a Voronoi cell repartitioning technique proposed in (Tewolde et al. 2008). We specifically use Algorithm 3 from (Tewolde et al. 2008) that takes a current Voronoi partition (a set of V_i 's) as an input and based on the load of the obstacle-free *poi*'s in each of the current Voronoi cells, a repartitioning is achieved. As a result, the participating robots are allocated to a unique and new polygonal region (possibly concave) in the environment for information collection. The algorithm virtually shifts the sites for the Voronoi partitions on the plane such that the new cells are balanced in terms of number of free *poi*'s in them and it terminates when the standard deviation of the load distribution is sufficiently low enough (Tewolde et al. 2008). Note that the repartitioning algorithm is only executed on the area bounded by the union of the Voronoi cells of the coordinating robots (\bar{R}_i). As the robots discover more obstacles in the environment, their perception about the environment will change and this will be reflected in their continuously repartitioned regions.

In case r_i and/or $r_j \in \bar{R}_i$'s *PoW*'s have changed after they last met (Case 2.a), they follow a similar strategy to repartition their current Voronoi cells as prescribed for case 1. When a robot reaches its path budget, i.e., covered B informative locations, it will stop exploring but continues to broadcast its *PoW* along with a *FINAL* message so that other robots recognize the stopped robot in the subsequent repartitions (lines 3-4 in Algorithm 1).

Theorem 1. *The region partitioning approach produces a complete coverage of the environment E .*

Proof. Coverage is complete when every $poi \in POI$ belongs to a Voronoi cell V_i of some robot r_i . From the initial partitioning, $\cup_n V_i = E \supseteq POI$. Upon a subset of robots \bar{R} meeting, it is the union of their current cell areas $\mathcal{V} = \cup_{r_k \in \bar{R}} V_k$ that is subject to repartitioning. The cell areas $\mathcal{V}' = E \setminus \mathcal{V}$ of all other robots $\mathcal{R} \setminus \bar{R}$ will not change. Because a Voronoi repartition also does not change \mathcal{V} (Tewolde et al. 2008), it remains true that $\mathcal{V}' \cup \mathcal{V} = E \supseteq POI$. \square

Conclusion

We have proposed a continuous Voronoi partitioning based informative path planning algorithm to collect maximal amount of information from an unknown environment using a set of mobile robots. Our approach is distributed in nature and each robot plans its local path only for one future informative way-point at a time. Whenever a set of robots come into each other's communication range, based on their currently discovered obstacles within their Voronoi cells and

corresponding free areas to explore, they repartition their cells for better load-balancing. In the future, we would thoroughly test the proposed approach within the Webots simulator and also using a team of TurtleBot 3 robots.

References

- Breitenmoser, A.; Schwager, M.; Metzger, J.-C.; Siegwart, R.; and Rus, D. 2010. Voronoi coverage of non-convex environments with a group of networked robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 4982–4989. IEEE.
- Cao, N.; Low, K. H.; and Dolan, J. M. 2013. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the 2013 International Conference on Autonomous agents and Multi-agent Systems*, 7–14.
- Choset, H., and Burdick, J. 1995. Sensor based planning, part i: The generalized voronoi graph. In *Proc. IEEE Int. Conf. on Robotics and Automation*.
- Guestrin, C.; Krause, A.; and Singh, A. P. 2005. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, 265–272. ACM.
- Hollinger, G., and Singh, S. 2010. Multi-robot coordination with periodic connectivity. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 4457–4462. IEEE.
- Hollinger, G. A., and Sukhatme, G. S. 2014. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research* 33(9):1271–1287.
- Hungerford, K.; Dasgupta, P.; and Guruprasad, K. 2016. A repartitioning algorithm to guarantee complete, non-overlapping planar coverage with multiple robots. In *Distributed Autonomous Robotic Systems*. Springer. 33–48.
- Karaman, S.; Walter, M. R.; Perez, A.; Frazzoli, E.; and Teller, S. 2011. Anytime motion planning using the rrt. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 1478–1483. IEEE.
- Kemna, S.; Rogers, J. G.; Nieto-Granda, C.; Young, S.; and Sukhatme, G. S. 2017. Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2124–2130. IEEE.
- Rasmussen, C. E., and Williams, C. K. 2006. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.
- Singh, A.; Krause, A.; Guestrin, C.; Kaiser, W. J.; and Batalin, M. A. 2007. Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, 2204–2211.
- Steven, F. 1987. A sweepline algorithm for voronoi diagrams. *Algorithmica* 2(1-4):153.
- Tewolde, G. S.; Wu, C.; Wang, Y.; and Sheng, W. 2008. Distributed multi-robot work load partition in manufacturing automation. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, 504–509. IEEE.
- Voronoi, G. 1907. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik* 133:97–178.