Multi-Robot Exploration

Sidney Stevens
Dept. of Mechanical Engineering
Stanford University
sidneyas@stanford.edu

Nishant Jannu
Dept. of Mechanical Engineering
Stanford University
nishant3@stanford.edu

Kyle Torres Casey
Dept. of Mechanical Engineering
Stanford University
kcasey2@stanford.edu

***Abstract*—This paper considers two distributed strategies for frontier-based multi-robot exploration: Non-Convex DisCoverage (NCD) and artificial potential fields (APF). We deploy simulated teams of N=4 agents to execute both algorithms to solve an exploration challenge in simulated environments. Our first algorithm is an adaptation of Non-convex DisCoverage (NCD), in which team of robots perform a local optimization to explore frontiers within their Voronoi cells. We adapt the original algorithm to use A* for path planning in non-convex regions. Our second algorithm extends traditional artificial potential field methods to include Voronoi frontier exploration, and A* for non-convex path planning. We compare the performance of NCD and APF in both uniform and an environment defined by a Gaussian density function. In both cases, artificial potential fields are able to fully explore the environment faster than Non-Convex DisCoverage, but we find that Non-Convex DisCoverage outperforms artificial potential fields at rapidly finding key features in the environment.**

***Index Terms*—multi-robot exploration, frontier methods, Voronoi, artificial potential fields, non-convex exploration**

I. Introduction

Ongoing developments in multi-robot exploration present vast potential for improvements in disaster response, environmental monitoring, and surveillance applications. While the domain remains largely research-based, recent natural disasters such as the 2021 Surfside Condo Collapse have demonstrated the potential for robots to assist search and rescue efforts, particularly when navigating through the wreckage of a structure is too dangerous or infeasible for first responders [1]. While such responses have largely relied on piloted teams of robots, the new frontier lies in the domain of autonomous search and rescue multi-agent operations. Ultimately, the goal of this research is to develop a deployment strategy for a team of robots to autonomously search a building and find humans in need of rescue. This paper considers the exploration problem for a homogeneous team of robots deployed within an unknown environment. We explore previous research that extends a Voronoi partition-based exploration strategy to a non-convex, bounded environment. The primary contributions of this paper include: 1) verifying the results of the previously developed convex and non-convex DisCoverage algorithms, 2) extending the underlying concepts and modifying the algorithm to rely on artificial potential fields rather than an orientation optimization, 3) comparing the performance of these algorithms, and 4) exploring the capability of the algorithms to perform density field estimation. We provide results from rounds of simulations to verify our results and provide the context for additional analysis.

II. Relevant Work

Multi-Robot exploration research focuses on how teams of robots can explore a new environment efficiently, often involving multi-robot SLAM. The DARPA Subterranean Challenge was a driving factor for recent research in this field. The primary objective of an exploration task is to cover the entire map in minimum time. Additional objectives include finding regions with the most important features, maximizing coverage and building out a robust map. Major research advancements have been made in the domain of multi-robot exploration over the past two decades.

A generalized exploration problem can be expressed as: given a set of m robots $R = \{r_1, r_2, ..., r_m\}$ deployed within a continuous and bounded environment, E, determine a sequences of poses for the robots $Q = Q_0, Q_1, ..., Q_n$ to detect features within the environment from on-board sensors [2]. [3] defines the borders between known (i.e. previously explored) and unknown regions of an environment as *frontiers*, where exploration includes a decision as to which frontier each robot should explore on the next iteration. Varying problems within the domain of multi-robot exploration have been proposed, including the *search problem* (which aims to maximize the detected targets in an environment) and the *coverage problem* (which aims to maximize coverage of an unknown region). In the classical formulation of the coverage problem, at each iteration the robots gain more knowledge of a previously unknown environment, and build a representation of the environment (e.g., a map) to make decisions as to where to move next to maximize their information gain. Often, the problem involves optimizing to reduce overall distance travelled [2] [4].

Exploration of a known or unknown environment in a multi-robot system introduces the inherent challenge of coordination among the team, particularly due to inherent and imposed limitations on communications between agents. To reduce the complexity of each individual robot's decision-making task, in frontier-based methods agents communicate only with neighbors in neighboring cells of a map of the environment, limiting the number of locations that each robot needs to consider. [3]

Some past research combines the topics of coverage control, which relates to finding configurations that allows the team

of robots to see the entire environment to multi-robot exploration. In [4], the authors show that a suitable modification of the coverage control objective to incorporate angular and distance information of the frontier can result in collision-free exploration of a convex region. This is done by iterative redefinition of the Voronoi regions and communication is restricted to Voronoi neighbours. They extend this work in [5] to support non-convex environments which include obstacles by performing a transformation of of the non-convex regions into star-shaped convex regions which allows application of the algorithm developed for convex regions. Further extensions use deep-reinforcement learning techniques to add collision avoidance for dynamic obstacles in the environment. [6] uses an imitation learning approach trained on human demonstrations for the control policy.

[7] is one of the most cited work in this field which proposed a method for task allocation and coordination among robots to explore unknown environments collaboratively. It made use of assigning utility values to unexplored areas and reduces these values when they come into the field of view of a deployed robot. An optimal navigation policy for each robot is arrived at by trading off between the utilities of unknown areas and the cost of reaching target points within them through dynamic programming. [8] expanded on this to incorporate communication interruptions, dynamic introduction and loss of robots while [9] added the ability to incorporate a known structure of the environment (such as partitions corresponding to rooms) into the task allocation process.

There has also been extensive research centered around finding important features or events while avoiding hazards. [10] addresses the problem of deploying multiple robots in an unknown environment. The robots only have access to a probabilistic model which they use to estimate the environment events and hazards online. The paper's key contribution is its novel use of mutual information. The robots use the history of previous robot failures while also trying to increase the "mutual information" of the group by following the gradient of mutual information.

The multi-robot exploration problem can also be set-up as a sequential decision-making process. [12] formulates this as a decentralised Markov-Decision-Process (Dec-MDP) which is a set of individual MDP's linked together via a distributed value function. More recent approaches like [13] employ spatial graph neural networks (SGNNs) by constructing a spatial graph representation of the environment to learn efficient coverage and exploration policies for the robots.

III. Problem Formulation

This project implements and compares the performance of two primary approaches to Voronoi-based coverage control in a non-convex environment: the non-convex DisCoverage algorithm proposed by [5], and a Voronoi-Based Artificial Potential Fields method. In addition, we implemented the convex DisCoverage algorithm proposed by [4], first in a convex environment and then in a non-convex environment to help understand the importance of the convex extension features in NDC. To set a baseline for comparison for all algorithms, we also implemented the *MinDist* algorithm proposed by [3], which determines an orientation to minimize travel distance by setting an agent's target point to the point on the frontier that is closest to the robot. The following sections will explain the major components of the convex DisCoverage, non-convex DisCoverage, and Voronoi-Based Artificial Potential Fields algorithms. We first consider the traditional Voronoi-based coverage problem:

$$\mathcal{H}_{\text{cover}}(\mathcal{P}) = \sum_{i=1}^{N} \mathcal{H}_{\text{cover},i}(\mathcal{P}) = \sum_{i=1}^{N} \int_{\mathcal{V}_i} f(q, p_i)\phi(q)dq \quad (1)$$

Where the robots divide their environment Q into n Voronoi cells such that:

$$\mathcal{V}_i = \{q \in Q \| q - p_i \| < \| q - p_j \| \forall j\} \quad (2)$$

For each agent i, every point in agent i's Voronoi cell is closer to agent i than to any other agent j. Each of the algorithms explored in this paper, DisCoverage and artificial potential fields, calculates the Voronoi field based on the traditional method.

A. Convex DisCoverage

The DisCoverage method utilizes the pseudo-algorithm illustrated in the following feedback loop:

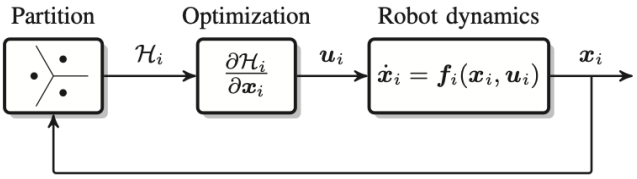

Fig. 1. Feedback Loop

An initial Voronoi partition is created based on the knowledge of the robot's initial locations and the known boundary of the convex space. Then, the objective function is optimized to find the new robot dynamics. After one iteration, the robots are in a new position and the process repeats. The novelty of this approach lies in the composition of the objective function, which modifies the equation for a traditional Voronoi coverage problem to only consider frontiers and determine an optimal orientation to maximize the density of frontiers in line with the current orientation. In the DisCoverage pseudo-algorithm: the robots first determine their Voronoi partitions by communicating with their Voronoi neighbors. Then, each robot, within their respective Voronoi cell, determines its optimal orientation to explore its visible frontiers, and subsequently updates its dynamics (position, $\dot{p}_i$) based on the optimal orientation.

$$\dot{p}_i = u_i = v \begin{pmatrix} \cos \delta_i^* \\ sin\delta_i^* \end{pmatrix} \quad (3)$$

We assume a constant speed for all robots, and update the position at each time step by calculating the optimal orientation δ_i^* for agent i, by:

$$\delta_i^* = \arg\max_{\delta_i} \mathcal{H}_{\text{discover},i}(p_i, \delta_i) \tag{4}$$

and the objective function $\mathcal{H}_{\text{discover}}$ considers the frontiers ∂S within the explored space S contained within the agent's Voronoi region:

$$\mathcal{H}_{\text{discover}}(\mathcal{P}, \Delta) = \sum_{i=1}^{N} \int_{\partial \mathcal{S}_i} f(p_i, \delta_i, q)\phi(q)dq. \tag{5}$$

with the performance function that multiplies the following two Gaussian distributions:

$$f(p_i)\delta_i, q) = \underbrace{\exp\left(-\frac{\alpha^2}{2\theta^2}\right)}_{\text{angular component}} \underbrace{\exp\left(-\frac{\|q - p_i\|_2^p}{2\sigma^2}\right)}_{\text{distance component}} \tag{6}$$

where θ and σ are hyper-parameters that weight the importance of the angle and distance (respectively) from the agent to its frontiers. A greater value of σ reduces the weight of far away frontiers, and a greater value of θ reduces the weight of frontiers oriented at a larger angle relative to the agent's current orientation.

The first Gaussian contains a variable α, which is defined as the difference in angle between the orientation of the robot (δ_i) and the frontier point (q) under consideration. Figure 2 displays the relationship between α and the robot environment.

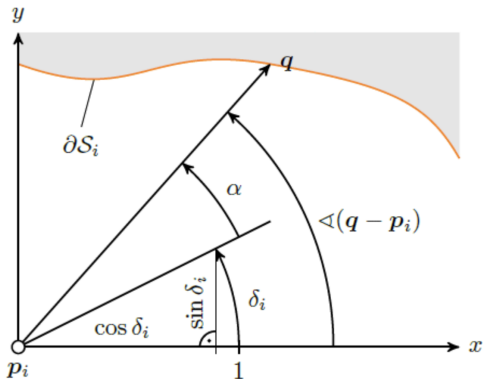

Fig. 2. Frontiers from robot field of view [5]

The variable α allows the performance function to minimize changes in angular displacement as well as euclidean distance. The change in behavior from centroidal search to exploring the Voronoi region is achieved by integrating the performance function along the frontier boundary only as opposed to the entire region.

B. Handling Non-Convexity: Non-Convex DisCoverage

The non-convex DisCoverage algorithm builds up the convex case to allow the controller to plan an optimal path within an environment that contains obstacles beyond other robots. The method handles non-convexity by transforming all unseen frontiers to the star-shaped domain [15], so that every previously discovered frontier is "visible" from the robot's point of view. The transformed objective function $\mathcal{H}^*_{\text{discover}}$ thus becomes:

$$\mathcal{H}^*_{\text{discover},i}(p_i, \delta) = \int_{\partial S_i^g} f(p_i, \delta_i, q^*)\phi(q^*)dq \tag{7}$$

where $q*$ is the transformed frontier, and the set of frontiers ∂S^g exists within the geodesic transformation of the explored space S^g. Visually, the algorithm implements the transformation as seen in 3 to enforce visibility of the frontiers, even as robots navigate around obstacles.

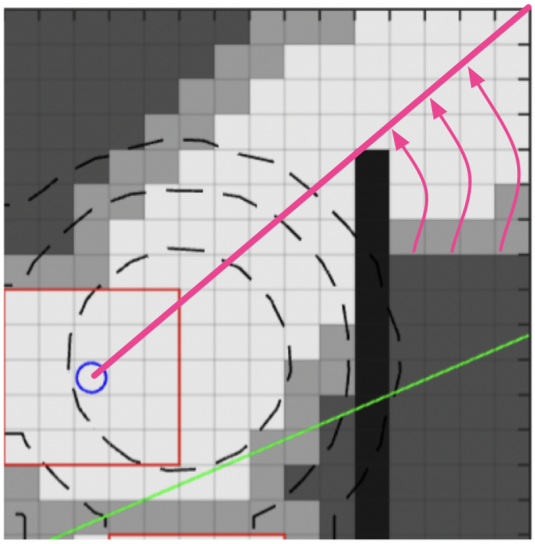

Fig. 3. Mapping Frontiers to Visible Regions

As shown above, all occluded frontiers in the robot's Voronoi region are mapped to a line at the boundary of the robot's field of view and the occluding obstacle. While [5] utilizes the Dijkstra algorithm to compute the shortest path and geodesic distance to the frontiers, we utilize A* search because it is a more efficient path planning algorithm that combines the advantages of heuristic search with Dijkstra [14]. Once we find the shortest path, we consider a ray in the direction of the first step of the path. Next, we map all the occluded frontiers to points on this ray that are equidistant to the robot as each frontier's original position. Through this transformation, we expect that the robot will move into a region where it can ultimately see these frontiers directly. It is worth mentioning here that a naive approach that ignores the occluded frontiers would fail if the occluded frontiers are the only ones left in a robot's Voronoi region. Hence, this transformation is pivotal to the algorithm's performance.

C. Voronoi-Based Artificial Potential Fields

One extension we considered incorporates the power of artificial potential fields into the Voronoi-based exploration methods outlined above, which use Voronoi regions to limit the frontiers for exploration. Whereas DisCoverage methods optimize for an orientation $\delta*$ that maximizes the visible frontiers along the robot's path of motion, Voronoi-Based Artificial Potential Fields apply virtual attractive forces to all frontiers within a given robot i's Voronoi cell, and virtual repulsive forces between robots in close proximity. As a result, we don't need an optimization step to find the optimal angle to move into as we implicity get it from the resultant force vector. The virtual forces F acting on an agent can be calculated as the gradient of a scalar potential field U:

$$F = -\nabla U \tag{8}$$

The scalar potential field U can be broken out into constituent components: $U = U_O + U_Q$, where U_O is the repulsive potential field the agent feels exerted by other agents in close proximity, and U_Q is the attractive potential field the agent feels from frontiers within its Voronoi cell. Thus the virtual forces can be similarly decomposed into: $F = F_O + F_Q$. The gradient of a scalar potential field is defined:

$$\nabla U(q) = \begin{cases} \eta\left(\frac{1}{R_0} - \frac{1}{D(q)}\right)\frac{1}{D^2(q)}\nabla D(q), & D(q) <= R_0 \\ 0, & D(q) > R_0 \end{cases} \tag{9}$$

Where R_0 defines the radius of relevance, and only objects (obstacles and frontiers) within this radius are considered. $D(q)$ defines the distance between the agent and the object being considered. η represents the weighted strength of the potential field. $\nabla D(q)$ points in the direction from the agent to the obstacle being considered. When calculating repulsive forces between agents, the value of η is a positive constant, i.e., $\eta = \eta_O$. When calculating attractive forces between an agent and its frontier cells, $\eta = \eta_Q$ is defined by:

$$\eta_Q = -k * exp(-\frac{\|q^* - p_i\|_2^2}{2\sigma^2})\phi(q*) \tag{10}$$

After obtaining the x and y components of the virtual force acting on an agent, F_y and F_x, we can obtain the new orientation for the agent:

$$\delta = \tan^{-1}(\frac{F_y}{F_x}) \tag{11}$$

A star-shaped transformation is still applied to frontiers obstructed behind obstacles, and agents still stay within their own Voronoi region. Thus, collision avoidance is still inherently incorporated into the algorithm. The additional repulsive forces between agents enable the robots to spread out more effectively throughout the environment rather than redundantly pursuing search spaces in close proximity simultaneously.

D. Estimating the Density Function

The previously discussed methods assumed that each agent knows apriori, the (non-uniform) density function within the environment to be explored. We assume that this density function is a mixture of Gaussians (Eq. 12) which is commonly used to express multi-modal data. In the exploration context, the Gaussians would represent important features in the map. For example, in the case of a search and rescue operation during a fire, it could represent the rooms of an apartment where people might be stuck or in the case of an earthquake, it can represent the amount of debris spread across the environment.

$$\phi(q) = \sum_{i=1}^{K} w_i.exp\left(-\frac{||q-c_i||^2}{2.\sigma_i^2}\right) \tag{12}$$

As an extension, we seek to relax the assumption that this density is known apriori and seek to estimate it in real time. We assume we can observe the feature density exactly at explored cells and use this data to fit the a mixture of Gaussians for each agent in a decentralised manner. Then the robots use this estimated density function to drive their exploration.

IV. Simulations

We consider a two-dimensional bounded world represented in a discretized occupancy grid as an arbitrary environment for our simulations as pictured below in Figure 4.

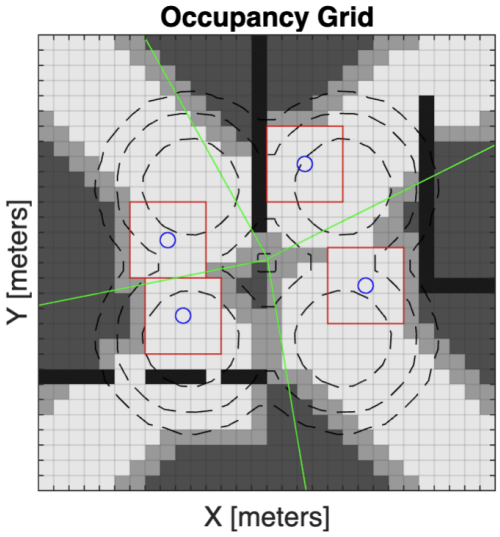

Fig. 4. Sample environment: explored (white), wall (black), unexplored (dark grey), frontier (light grey), density function (dashed), voronoi boundary (green) and lidar scan (red)

We have a 30×30 grid-world with 4 robots, each with a LiDar scan with a sensing depth of 2 cells. The dynamics of each robot is also discretized with 8 possible directions in which it can move (N, NE, E, SE, S, SW, W and NW). This discretization allows us to calculate the optimal orientation iteratively which is convenient as the performance function is non-convex. The density functions we consider in the simulations are either a mixture of 3 or 4 Gaussians. For

all of these parameters, we experimented with different values and only demonstrate results with the tuned values mentioned above. We run the simulations for 200 time steps or till all the cells are explored, whichever happens first.

As for conducting the simulations, we first considered only convex environments to replicate the results from [4]. Then, added obstacles to the map which makes the environment non-convex [5] and verified results on it. We use a discretized version of A* for finding the shortest path. Finally we demonstrate our two extensions: Voronoi-based APF and density function estimation. The density estimation uses the Expectation Maximization algorithm to do the fit. At each time step, the fit from the previous step is used to seed the fitting process.

V. Analysis

A. Non-Convex DisCoverage (NCD)

We include simulation results monitoring the evolution of four agents through a test environment for the NDC algorithm (see Figure 5). In the initial configuration, all four robots are positioned in the upper left-hand corner of the grid at what could be considered the entrance point of the environment. After 10 iterations, the robots have started exploring the space. Specifically, the purple and yellow robots have passed through the first Gaussian distribution. The reason why the other robots are delayed is because for the first few iterations, they have no frontier cells in their Voronoi region. This shows the importance of initial positions in Voronoi-based exploration. After 45 iterations, the yellow and purple robots have each discovered a Gaussian distribution (i.e., a key feature) positioned within a room. The orange robot briefly enters the same room as the yellow, but quickly realizes there are no frontiers left and leaves. After 104 iterations, the space is fully explored. The paths are fairly efficient, but not optimal.

Non-Convex DisCoverage shown in these figures is reliable in exploring the grid in a fast and efficient manner. The A* transformation described in Section II-B allows the robots to map and therefore explore out-of-view frontiers. After many iterations with different grids and initial positions, we found our results closely parallel the original DisCoverage paper [4]. Non-Convex DisCoverage clearly outperforms the *MinDist* algorithm in terms of area explored and density discovered at almost all time points (as seen in Figures 9 and 10). In Convex DisCoverage, they noted that uniform vs Gaussian densities had fairly similar run times. Not much on that topic was explored in the Non-Convex paper. Our data also showed that for Convex, the two run times were close. However, for Non-Convex there were slight variations in our results. We found that optimally located Gaussian distributions led to faster exploration times. In our figures, the grid with a Gaussian density function led to 23 fewer iterations than uniform density. A general pattern for uniform density emerged. On many occasions it spent too much time in the top left corner and it didn't spread out immediately like the grids with Gaussian distributions. However, the differences aren't substantially different.

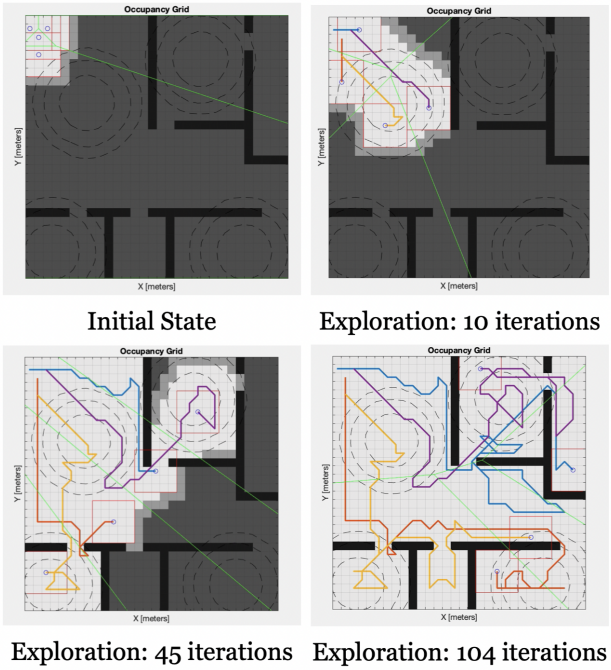

Fig. 5. Non-convex DisCoverage exploration over time

B. Artificial Potential Fields (APF)

In Figure 6, the robots start in the same location and in the same grid as Non-Convex Discoverage. Figure 6 shows that after 10 iterations, APF has covered roughly the same amount of space as DisCoverage. However, there are some notable differences in the paths the robots follow. The robot's now have repulsive potential fields amongst each other. So they don't follow paths close to each other and tend to push out to a wider configuration. The blue robot, in fact, doesn't even move forward because it is repelled back by the other three robots.

At 45 iterations, the amount of space explored is still similar to DisCoverage. The orange robot got stuck in the bottom left room because it was repelled by the yellow robot which was also in the room. Figure 6 shows that APF completely explores the grid within 85 iterations, which is 19 iterations faster than DisCoverage. One reason for this is because the purple robot enters the upper right room, whereas in DisCoverage the blue robot passes the purple to reach the last remaining frontier cells.

C. Performance: Non-Convex DisCoverage vs Artificial Potential Fields

APF proves to be effective in many different scenarios. Overall, regardless of LiDar depth, initial positions or density, APF consistently explores the entirity of the grid faster than DisCoverage. Figure 9 shows Area over Time for APF, Non-Convex DisCoverage, Convex DisCoverage and *MinDist*. The grid has a Gaussian density function and each robot has a LiDar depth of 2. We include *MinDist* in our plots to show how these algorithms improve a more basic strategy. *MinDist*

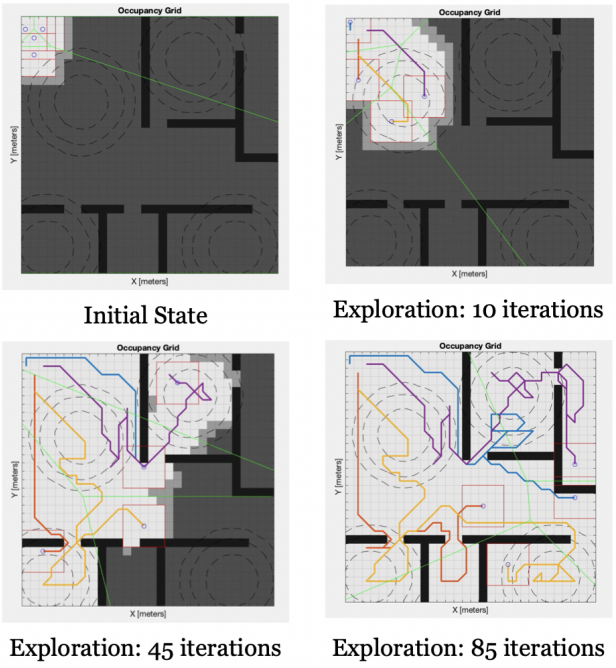

Fig. 6. Non-convex DisCoverage exploration over time

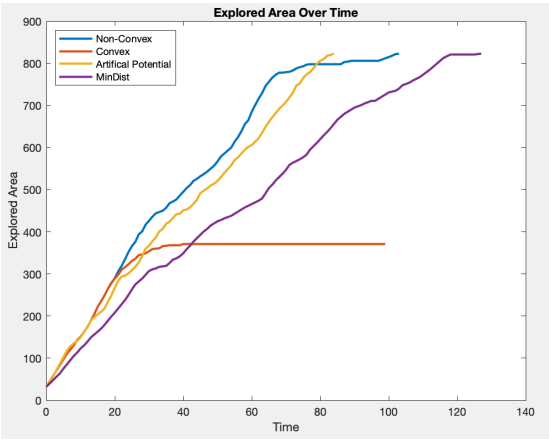

Fig. 7. Algorithms comparison: LiDar Depth: 2, Density: Gaussian

sends each robot to its closest frontier cell and doesn't take into account density or other variables. We also provide the Convex DisCoverage algorithm in the Non-Convex environment to illustrate the importance of the A* transform. In figure 9, NCD and APF clearly outperform Convex DisCoverage and MinDist. Although NCD seems better at first, APF is more steady in how much area it discovers and ultimately finishes first. However, figure 9 shows that NCD discovers the Gaussian distributions a couple iterations before APF. Although APF may finish faster overall, it is discovers all the key areas a little bit slower. However, if the LiDar depth is increased (e.g. depth = 3), APF not only finishes before NCD, but also discovers the Gaussians first as well. These exact trends are true for uniform density, too.

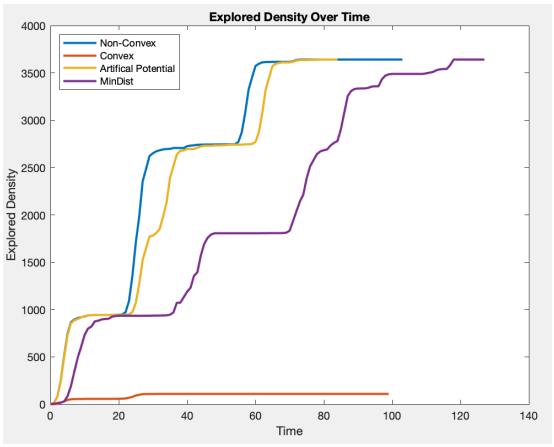

Fig. 8. Algorithms comparison: LiDar Depth: 2, Density: Gaussian

D. Performance: Estimation of Density function

When we use the estimated density function to drive our exploration, we finish the exploration faster but take longer to reach the important features as shown in the plots below. The estimation is able to fit only a subset of the underlying Gaussians (2 out of 3 in our case) as the center of the third Gaussian lies outside the map. This is a major pitfall as it can cause important features to be ignored or not receive priority.

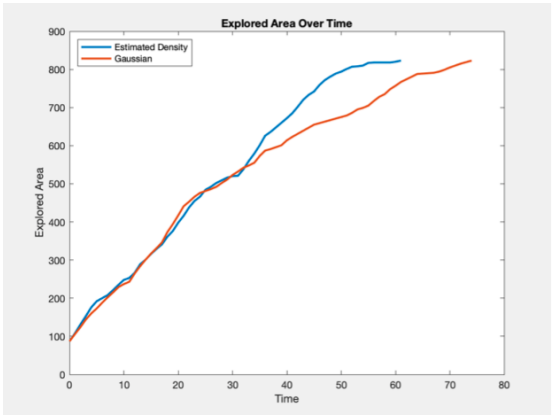

Fig. 9. Density estimation comparison: LiDar Depth: 2, Density: Gaussian

Addition of this density estimation component to the algorithm has a significant impact on the computational power required by the algorithm. As an improvement, we can consider that the agents share their fit parameters so that they can reach consensus on the parameters using a distributed neural network optimization technique [16] which can make the computation faster.

E. Algorithmic and Environmental Considerations

APF and NCD both provide similar results due to their many underlying similarities. The use of Voronoi partitions inherently avoids collisions. Additionally, Voronoi partitions work well in unknown environments, where the size of the environment is known but the specific layout is unknown.

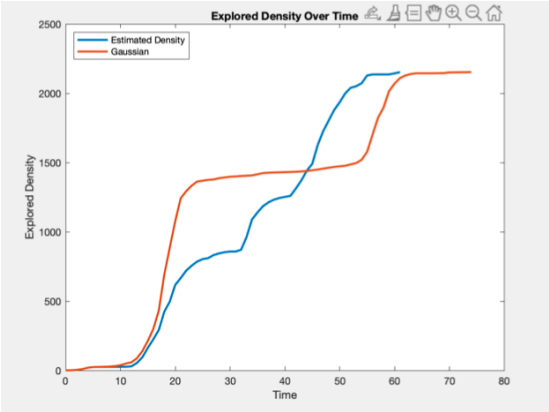

Fig. 10. Density estimation comparison: LiDar Depth: 2, Density: Gaussian

However, in a non-convex region, Voronoi partitions exhibit several weaknesses. For example, robots tend to get stuck in rooms in certain configurations because they can have a Voronoi region containing no frontiers. Additionally, if too many robots are collocated within a small space, certain robots may not move at all because they will never have a frontier within their Voronoi region. We were unable to find a way to prevent this behavior using NCD. However, APF could provide the opportunity to circumvent these challenges. One potential improvement would be to assign a weak attractive force to all frontiers, even those outside of a robot's Voronoi region, which would allow robots with no frontiers to move. This remains an area open for development in future work.

Another significant difference between APF and NCD lies in their relative quantities of hyperparameters. NCD only has two hyperparameters: σ and θ, which define the variances in the performance function. Due to the small number of hyperparameters, it is easier to tune for a given type of environment. However, in the current implementation APF has 7 hyperparameters (the coefficients of strength k_x and k_y, the radii of relevance R_O repulsive and R_O attractive, the weighted strengths of the potential field $\eta_O x$ and $\eta_O y$, and σ). This relatively large quantity of hyperparameters makes it significantly more challenging to discover optimal relationships between hyperparameters, even within a single environment. The potential field is defined by the following three hyperparameters: k_x, k_y, and R_O. For the repulsive fields between robots, k_x and k_y can be equal. However, for the fields created by frontier cells, the exploration is not optimal when they are equal. Also, k needs to be significantly bigger for the attractive fields in comparison to the repulsion by other robots. The radius of relevance for frontier cells ideally encompasses the entire grid so that frontiers aren't lost. The radius of relevance for the repulsive fields can be fairly small depending on the grid. The final hyperparameter is the σ that defines the variance in Equation 10. We found the optimal value to be roughly $\sigma = 4$, but this hyperparameter needs tuning in future work.

Many of the issues in our analysis came from bugs involving our discrete system. Assuming a discrete environment made the problem a lot simpler, but led to slightly inefficient motions. For example, both NCD and APF used discrete A* and this causes robots to get stuck in corners or in doors for small periods of time. APF, however, has more issues with the discrete formulation. If a robot enters a new undiscovered room perfectly symmetrically, all the frontiers will be equidistant from the robot and these attractive forces will cancel and lead the robot to oscillate between them. This more generally is an issue that applies to NCD and APF, but more consistently with APF.

VI. Future Work

We have presented novel extensions to multi-robot exploration using Voronoi partitions. Our work builds upon DisCoverage while also validating many of its results. Our main contribution for future work revolves around artificial potential fields (APF). APF mixed with Voronoi has many advantages over DisCoverage, but also has its drawbacks. A noticeable difference between the two methods centers around the number of hyperparameters. In future work, more testing and variations will help narrow down ideal hyperparameter values.

We wish to expand on the role of density functions. We want to iterate on the idea of changing density functions online. Static density functions aren't always realistic for certain environments. The ability to adapt to a changing environment could be a significant improvement. Another direction could be to create a probabilistic model for the density function where the density at a given position is stochastic.

Performance functions are the driving factor in the ability to explore the environment and we'd like to experiment with different objectives. In DisCoverage, they multiply an angular Gaussian and a distance Gaussian together to get their function. We want to incorporate the idea of energy efficiency of motion of the robots as well. This could be by minimizing the maximum distance covered by the robots and using this variable in our function. We could also explicitly prioritize the minimum time to reach certain events on the map.

The dynamics of the robot are assumed to be deterministic which can be relaxed. The motivation for this is that there might be certain areas of the environment which are harder to navigate in (imagining a muddy or slippery area) causing a change in the dynamic model of the robot. We can also expand upon the nature of the robot dynamics. Instead of controlling the orientation with a constant velocity, we could allow for variable velocities and more complex actions.

VII. Acknowledgements

We thank the course staff of AA277 - the instructor, Prof. Mac Schwager and the course assistants, Joe Vincent and Trevor Halstead, for their continued support and feedback for the project.

REFERENCES

[1] R. Murphy, "How Robots Helped Out After the Surfside Condo Collapse," 2021, IEEE Spectrum.

[2] A. Quattrini Li, "Exploration and mapping with groups of robots: recent trends", Curr Robot Rep 1, 227–237 (2020).

[3] B. Yamauchi, "Frontier-Based Exploration Using Multiple Robots," Proc. AGENTS, 1998, pp. 47–53.

[4] A. D. Haumann, K. D. Listmann and V. Willert, "DisCoverage: A new paradigm for multi-robot exploration," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 2010, pp. 929-934, doi: 10.1109/ROBOT.2010.5509993.

[5] D. Haumann, A. Breitenmoser, V. Willert, K. Listmann and R. Siegwart, "DisCoverage for non-convex environments with arbitrary obstacles," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 4486-4491, doi: 10.1109/ICRA.2011.5980415.

[6] J. Hu, H. Niu, J. Carrasco, B. Lennox and F. Arvin, "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14413-14423, Dec. 2020, doi: 10.1109/TVT.2020.30348

[7] W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, "Collaborative multi-robot exploration," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 476-481 vol.1, doi: 10.1109/ROBOT.2000.844100.

[8] 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), Washington, DC, USA, 2002, pp. 3016-3023 vol.3, doi: 10.1109/ROBOT.2002.1013690.

[9] K. M. Wurm, C. Stachniss and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 2008, pp. 1160-1165, doi: 10.1109/IROS.2008.4650734.

[10] Schwager, M., Dames, P., Rus, D., Kumar, V. (2017). A Multi-robot Control Policy for Information Gathering in the Presence of Unknown Hazards. In: Christensen, H., Khatib, O. (eds) Robotics Research . Springer Tracts in Advanced Robotics, vol 100. Springer, Cham.

[11] RR. Murphy, S. Tadokoro, A. Kleiner, "Disaster robotics." In: Springer Handbook of Robotics. Springer; 2016. p. 1577–604. https://spectrum.ieee.org/building-collapse-surfside-robots

[12] L. Matignon, L. Jeanpierre and A. -I. Mouaddib, "Distributed value functions for multi-robot exploration," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 2012, pp. 1544-1550, doi: 10.1109/ICRA.2012.6224937.

[13] Ekaterina Tolstaya, James Paulos, Vijay Kumar, and Alejandro Ribeiro. 2021. Multi-Robot Coverage and Exploration using Spatial Graph Neural Networks. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE Press, 8944–8950.

[14] P. Premakumar (2023). "A* (A Star) search for path planning tutorial." https://www.mathworks.com/matlabcentral/fileexchange/26248-a-a-star-search-for-path-planning-tutorial

[15] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in Proc. of the 47th IEEE Conference on Decision and Control, 2008.

[16] Yu, Javier et al. "DiNNO: Distributed Neural Network Optimization for Multi-Robot Collaborative Learning." IEEE Robotics and Automation Letters PP (2021): 1-1.