Visibility-Voronoi Algorithms With Unknown Environments and Time-Varying Density Fields

Victor Qin[1]

***Abstract*— Two motivating problems in multi-agent robotics are the exploration problem of discovering an unknown environment and the observation problem of maintaining as much visibility over the environment. Together, they can be formulated as a version of the deployment problem, where we must try to evenly distribute agents through an environment while mainting observation of the environment. Visibility-Voronoi algorithms give a realistic and tractable solution - in this paper, I extend an algorithm given by Kantars et al. towards unknown environments. In addition, I analyze and show promising results for the use of time-varying density functions in influencing robot behavior.**

I. INTRODUCTION

The advent and proliferation of mobile robotics has made it possible for multi-agent robotic systems to perform a wide variety of tasks, including exploration, observation, and coordinated motion. These different mechanisms have significant impacts in wide-ranging fields from disaster response to environmental monitoring and surveillance. Two well-studied fields of multi-agent robotics include the exploration problem and and the observation problem. Multi-agent exploration requires the coordination of a robotic system through an unknown environment, usually by coordinating through algorithms like Simultaneous Localization and Mapping (SLAM). In contrast, the observation problem looks to distribute agents in a way as to maximize the area of the environment observed by the system. This is commonly done using variants of Voronoi partitioning, by drawing equidistant lines between neighboring agents.

A combination of these two questions forms a version of the deployment problem - given a set of agents, how do we evenly distribute them in an unknown environment while maintaining observation of the environment? This was partly answered in Kantaros, Thanou and Tzes (2015) [1], in their formulation of Visibility-based Voronoi diagrams (VbV). They present a gradient-descent control law for robotic agents based on an agent's visible environmental area, assuming perfect knowledge of the environment and maintaining communication links with other agents.

Taken one-step further, we can further imagine patrol or dynamic monitoring systems built off of work from deployment algorithms. This is achieved through time-varying environmental density functions, which can influence robot positioning and movement over time.

[1]Victor Qin is an undergraduate in Electrical Engineering at Harvard University. `victor_l_qin@college.harvard.edu`

Special thanks to Yiannis Kantaros, postodoctoral researcher at University of Pennsylvania

I will first cover the Visibility-based Voronoi partitioning method presented in [1], then introduce my notation for operating the algorithm in unknown environments. The limitations with such an algorithm motivates the use of a time-varying reward density function for the environment, and generates interesting patrol behavior for multi-agent systems. Finally, I will present simulation results for the system, and identify promising next steps.

II. DEPLOYMENT PROBLEM

First, I will go over the formulation for the generic deployment problem, which aims to spread a multi-robot system across an unknown environment. We assume the robots operate in a compact, non-convex environment $A \in \mathbb{R}^2$. A density function $\phi(q)$ measures the reward for monitoring any point $q \in A$. We define the boundary of any polygon as $\delta(\cdot)$, such that the boundary of A is represented by $\delta(A)$. Because we assume the system does not have full awareness of the environment, we define the robot system's knowledge of the boundaries of A at any moment in time as $\delta(A,t)$. This time-varying knowledge of the boundary will influence the response of robots.

A system of n robots monitors A, where the positions are represented by $X(t) = \{x_1(t), x_2(t), \ldots, x_n(t)\} \in A$ and $x_i(t) \in \mathbb{R}^2$, and each robot i has a sensor detection range of r_i. We use a very simple movement model - that is, we define the movement of robots as:

$$x_i(t+1) = x_i(t) + u_i(t), \quad u_i \in \mathbb{R}^2,\ \forall i = \{1, \ldots, n\} \tag{1}$$

This system has a sensor model based on a 360-degree angle laser rangefinder. Assume that each robot has a sensor range limit r_i and takes k measurements such that each measurement is separated by $360/k$ degrees. The measurements $s_i(x_i, \omega)$ where ω is the orientation of the robot will be:

$$s_i(x_i,\omega) = \begin{cases} r_i & [x_i,] \notin A \\ \|q_i^A - x_i\| & [x_i, x_i + r_i(\cos(\omega), \sin(\omega))] \in A \end{cases} \tag{2}$$

where $x_i^A = x_i + r_i(\cos(\omega), \sin(\omega))$ is the endpoint of a ray from x_i in the direction of ω with length r_i, and $q_i^A = [x_i, x_i + r_i(\cos(\omega), \sin(\omega))] \cap \delta(A)$ is the intersection of the ray with the boundary $\delta(A)$.

In addition, we assume that the robots are unlimited in communication range and bandwidth.

III. VISIBILITY-BASED VORONOI DIAGRAMS

In this section I will review the VbV formulation given in [1]. This is important for understanding the gradient descent method that Kantaros presents, and will motivate our modifications to the density function.

We will first define the set of visible points from robot i as VP^i, where:

$$VP^i = \{q \in A | [x_i, q] \in A\} \tag{3}$$

$[x_i, q]$ is the line segment between an arbitrary point q and the location x_i. VP^i is thus the set of all points where $[x_i, q]$ does not intersect $\delta(A)$.

Then, we define the "visibility disc" C^i as all points visible from robot i and within radius r_i:

$$C^i = \{q \in VP^i \cap B^i\} \tag{4}$$

where $B^i = \{q | \|q - x_i\| \leq r_i\}$.

This effectively models the set of points visible to an omni-directional sensor in an arbitrary closed environment, whether convex or non-convex. This work is based on Defn. 1 and 2 in [2].

Next, we can quickly introduce the concept of Voronoi diagrams. In general, for points $x_1, \ldots x_n \in B$, where B is a convex and compact environment, the Voronoi polygon V_i for point x_i is defined as:

$$V_i = \{q \in A | \|q - x_i\| \leq \|q - x_j\| \forall j \in 1, \ldots n \neq i\} \tag{5}$$

In essence, the Voronoi polygon for a point x_i is the set of all points closer to x_i than any other point x_j. Our formulation of the non-convex deployment problem requires two modifications to this definition: first, we are considering heterogenous sensor radii r_i, which complicates the definition of distance; and second, the non-convexity of our environment requires us to incorporate the concept of visibility polygons introduced above.

To address the first issue, the authors in [2] use a power diagram to divide points between different agents. In Defn. 3, they define the Voronoi power diagram V^i_{pd} for robot i as:

$$V^i_{pd} = \{q \in A | \, \|q - x_i\|^2 - r_i^2 \leq \|q - x_j\|^2 - r_j^2 \, \forall j \neq i\} \tag{6}$$

We can then incorporate the concept of range-limited visibility polygons by using our definition of C_i set earlier. We thus define the range-limited Voronoi polygon $V^i_{r,pd}$ as the intersection between the Voronoi power diagram and our range-limited polygon:

$$V^i_{r,pd} = \{q \in V^i_{pd} \cap C_i\} \tag{7}$$

This is illustrated best in Fig. 1 of [2].

IV. MOTION CONTROL

In this section, I'll discuss the control law we'll use to move the robot around the environment. In addition, we can demonstrate that the control law presented in [1] and [2] also consistently works when the robot explores an unknown boundary.

A. Control Law

Kantaros et al. propose a gradient based control law in [1] and [2] based on the visibility boundaries of a robot. We define 4 types of boundaries:

1) A robot's boundary with the environment $\delta(A)$
2) A robot's visibility boundary with another robot
3) The range-limited boundary, where the robot's view is limited by the range of the sensor
4) The node-limited boundary, where the robot's view is limited by a non-convex corner of the environment

Of those boundary types, only the last two serve as indications of possible directions of exploration for the robot. They are indicators of empty space for the system to expand into and observe. We can move robots in a decentralized fashion by observing the density function along those boundaries to estimate the value gained from motion in any one direction.

The node-limited boundary is influenced by several vertex points of the range-limited Voronoi polygon. Because it tries to push the robot in directions that can see more of the space around a non-convex point, it lies along the line from x_i to a non-convex point. We will define $p^i_{b,l}$ as the farther point of a node-limited boundary, $p^i_{a,l}$ as the near point, and $p^i_{r,l}$ as the non-convex corner defining this boundary. See Fig. 5 in [2] for an image description of the space.

Thus, the proposed control law is:

$$u_i = \int_{\delta(V^i_{pd} \cap \delta(B^i)} \phi(q) n^i_0(q) dq + \sum_{l=1,\ldots,n} \left(\frac{n^i_0}{\|p^i_{r,l} - x_i\|} \int_{\delta^l_{min}}^{\delta^l_{max}} \phi(Q) \delta d\delta \right)$$

where $Q = p^i_{r,l} + \delta \frac{p^i_{r,l} - x_i}{p^i_{r,l} - x_i}$, $\delta^l_m in = \|p^i_a - p^i_{r,l}\|$, and $\delta^l_m ax = \|p^i_b - p^i_{r,l}\|$.

B. Dealing With Unknown Environments

We can notice that the control law proposed above only requires knowledge of the local boundaries around the robot, including nonconvex corners of the environment. The control law will still hold even if we base it on a robot's own estimate of the environmental boundaries.

Assume the system shares an occupancy map $A^{est} = \{a_1, \ldots a_m\}$ that represents points on $\delta(A)$ that have been viewed by robots $x_i \in X$, where $a_j = 1$ if observed as occupied by a robot and 0 otherwise. We can get a reasonably accurate model for A from A^{est} by interpolating between points in A^{est}. The 4 boundary types listed above can be derived from A^{est} by the following methods:

1) Environmental boundary. Let the maximum distance between points in A^{est} be $\epsilon = 360/k \cdot r_i$ (a function of the robot sensor's resolution), and let the smallest width between two walls in A be w_{min}.
We build our interpolation as follows - for points $s_i(x_i, \omega_j), s_i(x_i, \omega_{j+1})$, we identify the line $[s_i(x_i, \omega_j), s_i(x_i, \omega_{j+1})]$ as an estimate for $\delta(A)$ if $\|s_i(x_i, \omega_j) - s_i(x_i, \omega_{j+1})\| \leq epsilon$. Our major worry is that a false boundary connecting across the interior of A will be constructed. The estimation proposed holds as an accurate estimate as long as $w_{min} > \epsilon$ - if the smallest distance between opposite walls of A is larger than the worst resolution, we don't have to worry about false boundaries being constructed.
2) Visibility polygon boundaries. These boundaries between two adjacent visilbility polygons happen in free space and are not directly dependent on A.
3) The range-limited boundary. Again, this also happens in free space and is not directly dependent on A.
4) The node-limited boundary. This depends on the non-convex corners of A, which we must somehow identify from A^{est}. We can do this by looking for consecutive measurements $s_i(x_i, \omega_j), s_i(x_i, \omega_{j+1})$ where $s_i(x_i, \omega_j) = r_i$ and $s_i(x_i, \omega_{j+1}) \leq r_i$, and $[s_i(x_i, \omega_j), s_i(x_i, \omega_{j+1})] \notin \delta(A)$.
We can note that if $\|s_i(x_i, \omega_j) - s_i(x_i, \omega_{j+1})\| > epsilon$, then it is likely that the points are not in $\delta(A)$. Then, we can construct a node-limited boundary by taking the boundary defined by $\delta(A)$ and taking the non-convex node at the point where $[x_i, s_i(x_i, \omega_j)] = \delta(A)$. This construction can be estimated using other measured values, eg. $[s_i(x_i, \omega_{j+1}), s_i(x_i, \omega_{j+2})] \in A$.

Given these methods for identifying the boundaries used in Kantaros et al. for calculating the gradient, we can calculate the control law given in (8) in unknown environments.

V. Time-Varying Density Fields

One potential problem with the control law proposed in [1] and [2] is the slower exploration speed of the environment. If a robot is not touching any adjacent robot's range-limited Voronoi polygon, there is no differential gradient pushing the robot towards a direction. Two forces as a result of range-limited boundaries on opposite sides of a robot's Voronoi polygon will cancel out (pointing in opposite directions), so a robot can end up stuck in local minima and unmoving if other robots don't interact with it and disrupt its range-limited boundary. This is in fact how the algorithm presented in [2] is shown to terminate if the total area coverage of the robot's sensors is less than the area of the environment (see Fig. 10 in [2]).

To overcome this fact, I experimented with time-varying density functions $\phi(t, q) = [1, 0]$. In general, time-varying density functions play a role as follows:

1) Every point $q \in A$ in the environment begins with $\phi(0, q) = 1$.
2) If a point $q \in A$ is not visible from any robot, it "regenerates" density at time t:
$$\phi(t+1, q) = \phi(t, q) + \alpha \cdot \max(0, \beta - \phi(t, q)) \quad (8)$$
where $\alpha \in (0, 1), \beta \in (0, 1]$. This is an exponentially regenerating function. If $\phi(t_0, q) = 0$, then:
$$\phi(t_0 + t, q) = \beta - e^{-\alpha\beta t} \quad (9)$$
This equation caps the regenerated value at β.
3) If a point q is within a robot's range-limited Voronoi polygon $V^i_{r,pd}$ at time t, its density $\phi(t+1, q) = 0$.
4) This density distribution is then used to calculate the control law, and we repeat Steps 2-3.

This density law encourages exploration by robots on the frontier of the explored area $\delta(A, t)$, yet will still pull following robots to spread out around A.

Even more notably, this regeneration effect encourages robots to revist points and defines non-deterministic patrol behavior from the system of robots. While I have not theoretical results, it seems that given enough robots the system will tend towards robots patrolling certain defined sectors of the environment. This can be seen better in the simulation results given in the next section.

VI. Simulation Results

I implemented and ran the control law presented in [2] with both static and time-varying density fields in unknown environments in MATLAB. I tested it on a system of 10 robots, starting in two locations, and used the same system for both static and time-varying tests.

A. Static

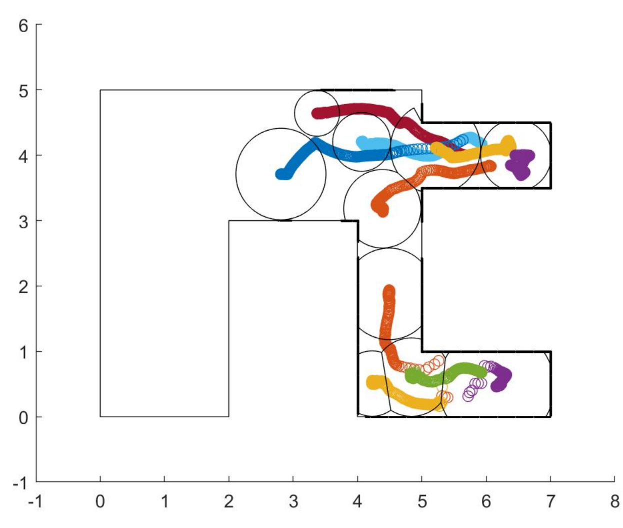

Fig. 1. Static density function, ran for 433 iterations. System finds local minima at this point and stops progressing

The static simulation results are presented in Fig. 1. Each robot is shown in a different color, with a trail marking its path taken. The black marks along the border of the environment show points along $\delta(A)$ that have been identified and viewed. Because each robot has found a local minimum, where the gradient forces in every direction cancel out, the system stops at this point.

B. Time-Varying

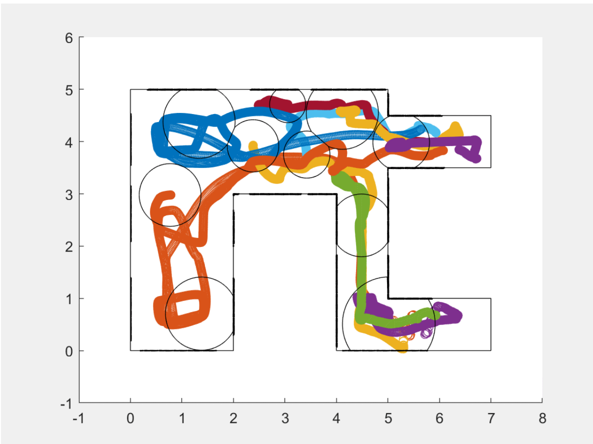

Fig. 2. Time-varying density function, ran for 1000 iterations. Patterns begin to emerge in agent actions.

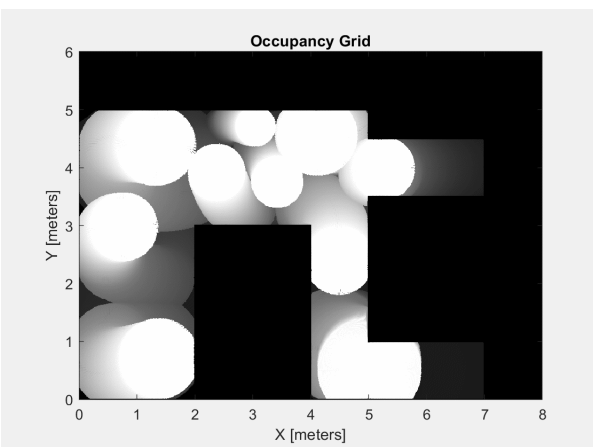

Fig. 3. Visibility map, showing the state of the visibility density after 1000 iterations

Fig. 2 shows the paths taken by 10 robots over 1000 iterations in the same environment and starting positions presented in the static case. The border is less clearly marked due to some restarts while running the simulation. Fig. 3 visulaizes the time-varying density function, with darker areas closer to 1 and lighter areas closer to 0. Each robot is tracing out a "trail" in the density function that encourages other robots to explore around rather than retracing. Animated gifs of the simulation can be found at: https://imgur.com/a/6mBaNhY.

This method was effective in more quickly exploring the environement, as the leading robot quickly jumped ahead as a result of the density differential artificially generated by setting explored points to zero density. Patterns of patrol begin to emerge, as robots balance out the areas they monitor.

VII. REFLECTIONS

I had a lot of fun working on simulations and building out the project, although a tight schedule prevented me from further developing some points. A lot of the ideas from gradient descent and objective functions were useful in understanding some aspects of this project. I would have enjoyed some discussions of multi-agent systems or maybe even multi-arm path planning, but those are minor considerations compared to the large amount of robotics knowledge ES 159 taught.

VIII. CONCLUSIONS

There are many more promising applications of time-varying density functions in order to control multi-agent systems. This paper highlights promising avenues of theoretical research in time-varying functions, as well as extending and proving some of the results from Kantaros et al. in unknown environments.

REFERENCES

[1] Kantaros, Yiannis, Michalis Thanou, and Anthony Tzes. "Visibility-oriented coverage control of mobile robotic networks on non-convex regions." 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014.

[2] Kantaros, Y., Thanou, M., amp; Tzes, A. (2015). Distributed coverage control for concave areas by a heterogeneous Robot–Swarm with visibility sensing constraints. Automatica, 53, 195-207. doi:10.1016/j.automatica.2014.12.034

[3] Diaz-Mercado, Yancy, Sung G. Lee, and Magnus Egerstedt. "Distributed dynamic density coverage for human-swarm interactions." 2015 American Control Conference (ACC). IEEE, 2015.