Multi-Robot Coverage Control from Time-invariant Density Map

Shall Chee Shih, Jiatong Sun, Muxuan Wang

University of Pennsylvania

Abstract:
The paper discusses the study of multi-robot coverage control, introduces and implements the Lloyd algorithm based on Voronoi tessellation, proposes and simulates a series of optimizations and analyzes the results. By running a decentralized controller in time-invariant density function, the algorithm allows the team of agents to achieve a centroidal Voronoi tessellation. The effectiveness of control policy is proven by simulation in two representative density maps.

Index Terms: Multi-Robot, Coverage Control, Voronoi Tessellation, Decentralized, PD Control

1. Introduction

The fast development of swarm robots has been utilized in a lot of fields, and as electromagnetic systems advances, robot agent size decreases, sensor accuracy sharpens and computation power bursts, implementations of robots in large numbers are stretching into more demanding tasks. Examples of such systems include Autonomous Oceanographic Sampling Network, in which robots are required to measure temperatures, currents and other oceanographic signals, and each robot is equipped with acoustic local area network and coordinate to generate global data. Common heuristic approaches, which are extensively investigated, would propose a control policy to drive each robot simultaneously given a predefined environment. In such aforementioned system, however, traditional control policy no longer applies, as robots are envisioned to dwell in largely grey environments, with communication only locally viable, centralized control that governs the whole swarms would prove infeasible; a decentralized control that copes with adaptive environment with asymptotically convergent robustness has thus become a necessity. In our study, we implement a series of such distributed algorithms to simulate the task of returning a number of robots to a given density map, and discuss the effect of optimizations upon former studies.

2. Coverage Control with Time-invariant Density Function

The ultimate goal of the project is to compute the target locations of inter-mediated relay agents that ensure a team of robots stays in contact with each other. The grander project involves two parts: the first part is proposing and implementing a data-driven supervised learning model to optimize the algebraic connectivity of a team of agents, which uses a set of algebraic connectivity optimization schemes of the target network agents' locations given the task agents' locations as the training set. After training, the model is used to predict the probability of the target network agents' locations given a certain agents' locations and generate a density map to address probability distribution at each location. The second part is to calculate the exact target network agents' locations from the density map.

The first step is currently being studied by another group in Vijay Kumar Lab; this paper focuses mainly on the second step, as will be elaborated in following sections.

2.1. Coverage Control

Many multi-robot systems such as exploration robots involve controlling a team of agents from one configuration to another. This is the case for coverage control problems, where agents are tasked to distribute themselves in an environment. There are many applications for coverage control, such as surveillance of an area in the filed by a team of robots. While a common approach involves computing control actions for each individual agent, this strategy does not scale well as the number of agents grows. Recent work seeks to solve this problem by treating a multi-agent team as a distribution and controlling the distribution rather than individual agents.

2.2. Voronoi Partitions

The positions of a team of robots could be denoted by $p_i \in \mathbb{R}$, $i \in \{1, ..., N\}$, in which d = 2 for two-dimensional planner task, d = 3 for three-dimensional aerial task. The density function is the function that encodes the relative importance of each point q in the domain $D \subset \mathbb{R}$, which is the probability of the target network agents' locations in this paper. The density function ϕ maps the points in the domain D to $[0, \infty)$, where $\phi(q) = 0$ for most points in this paper. In order to calculate the performance of certain distributions, it is a common choice to divide the domain into several cells by using Voronoi cell, which is defined by

$$V_i(p) = \{q \in D | \, \|q - p_i\| \leqslant \|q - p_j\|, \forall j \in \mathcal{N}\} \tag{1}$$

in which robot i,$i \in \{1, ..., N\} := \mathcal{N}$, is in charge of points closest to it.
The quality of the coverage is encoded by the cost function [1]

$$\mathcal{H}(p) = \sum_{i=1}^{N} \int_{V_i(p)} \|p_i - q\|^2 \phi(q) dq \tag{2}$$

which is the locational function. Lower loss function indicates better performance of the coverage.

2.3. Centroidal Voronoi Partitions

Centroidal Voronoi Partitions(CVT) means each robot i are at the mass center of its correspondent Voronoi cell. The mass center of each Voronoi cell i is defined as

$$C_{V_i}(p) = \frac{\int_{V_i(p)} q\phi(q)dq}{\int_{V_i(p)} \phi(q)dq} \tag{3}$$

Then, by using the parallel axis theorem, the minimized cost function could be written as

$$\mathcal{H}_V(P) = \sum_{i=1}^{N} \int_{V_i(p)} (\|q - C_{V_i}\|^2 + \|p_i - C_{V_i}\|^2)\phi(q)dq \tag{4}$$

Let

$$J_{V_i, C_{V_i}} = \int_{V_i(p)} \|q - C_{V_i}\|^2 \phi(q)dq \tag{5}$$

then equation (4) can be rewritten as

$$\mathcal{H}_v(P) = \sum_{i=1}^{N} J_{V_i, C_{V_i}} + \sum_{i=1}^{N} M_{V_i} \|p_i - C_{V_i}\|^2 \tag{6}$$

Assume $\frac{\partial C_{V_i}}{\partial p_i} = 0$, then the partial derivative of $\mathcal{H}_v$ is that

$$\frac{\partial \mathcal{H}_v}{\partial p_i}(P) = 2M_{V_i}(p_i - C_{V_i}) \tag{7}$$

Therefore, the cost function reaches minimum only when each robot i is at the mass center of its correspondent Voronoi cell, i.e., in CVT.

3. Vehicle Dynamics

3.1. Control for Linear System

Assume the system follows a first-order dynamical behavior:

$$\dot{p}_i = u_i \tag{8}$$

In order to minimize $\mathcal{H}_v$, we could control each robot follow a gradient decent. Therefore, we set the input as

$$u_i = -k_p M_{V_i}(p_i - C_{V_i}) \tag{9}$$

in which k_p is a positive gain. By setting $\mathcal{H}$ as the Lyapunov function and using LaSalle's principle, we could prove that the robot position converges to the centroidal Voronoi configurations [1].

3.2. Control for Nonlinear System

In this section, we extend from the linear system to nonlinear system. For the devised proportional-derivative (PD) control:

$$u_i = -k_p M_{V_i}(p_i - C_{V_i}) - k_d \dot{p}_i \tag{10}$$

where‘ k_d is a positive gain. We can analyse the closed-loop system induced by this control law with the following Lyapunov function:

$$V = \frac{1}{2} k_p \mathcal{H}_V + \frac{1}{2} \sum_{i=1}^{N} \dot{p_i}^2 \tag{11}$$

We prove its asymptotic stability by following steps:

1. At $P_0 = \{p_{i0} = C_{V_i}, i \in \{1, 2, \cdots, N\}\}$, all robots have reached their equilibrium, indicating $\dot{p}_{i0} = 0, \forall i \in \{1, 2, \cdots, N\}$. Substituting into the Lyapunov function, we have

$$\begin{aligned} V(P_0) &= \frac{1}{2} k_p \mathcal{H}_V(P_0) + \frac{1}{2} \sum_{i=1}^{N} \dot{p}_{i0}^2 \\ &= \frac{1}{2} k_p \sum_{i=1}^{N} \int_{V_i} \|p_{i0} - q\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=1}^{N} \dot{p}_{i0}^2 \\ &= \frac{1}{2} k_p \times 0 + \frac{1}{2} \times 0 = 0 \end{aligned} \tag{12}$$

   And for $\forall P = p_i \neq P_0$, since $k_p > 0$, $\mathcal{H}_V > 0$ and $\sum_{i=1}^{N} \dot{p}_{i0}^2 > 0$, we have $V(P) > 0$. This gives

$$\begin{cases} V(P) = 0, & P = P_0 = \{p_{i0} = C_{V_i}, i \in \{1, 2, \cdots, N\}\} \\ V(P) > 0, & otherwise \end{cases} \tag{13}$$

2. The time derivative of Lyapunov function is

$$\begin{aligned} \frac{d}{dt} V &= \frac{1}{2} k_p \frac{d}{dt} \mathcal{H}_V + \frac{1}{2} \sum_{i=1}^{N} \frac{d}{dt} (\dot{p_i}^2) \\ &\leqslant k_p \sum_{i=1}^{N} M_{V_i}(p_i - C_{V_i}) \dot{p}_i + \sum_{i=1}^{N} \dot{p}_i u_i \\ &= k_p \sum_{i=1}^{N} M_{V_i}(p_i - C_{V_i}) \dot{p}_i + \sum_{i=1}^{N} \dot{p}_i (-k_p M_{V_i}(p_i - C_{V_i}) - k_d \dot{p}_i) \\ &= -k_d \sum_{i=1}^{N} \dot{p_i}^2 \leqslant 0 \end{aligned} \tag{14}$$

The equality only holds when $\dot{p}_i = 0$, or $p_i = C_{V_i}$. This gives

$$\begin{cases} \dot{V}(P) = 0, & P = P_0 = \{p_{i0} = C_{V_i}, i \in \{1, 2, \cdots, N\}\} \\ \dot{V}(P) < 0, & otherwise \end{cases} \tag{15}$$

3. When $\|p_i - C_{V_i}\| \to \infty$, i.e. robots deviate from their Voronoi tessellation centroids, we have

$$\begin{aligned} \mathcal{H}_V &= \sum_{i=1}^{N} \int_{V_i} \|p_i - q\|^2 \phi(q) dq \\ &= \sum_{i=1}^{N} J_{V_i, C_{V_i}} + \sum_{i=1}^{N} M_{V_i} \|p_i - C_{V_i}\|^2 \\ &\to \infty \end{aligned} \tag{16}$$

since $k_p > 0$, $\frac{1}{2}\sum_{i=1}^{N} \dot{p_i}^2 > 0$, thus $V(P) \to \infty$, the Lyapunov function is radially unbounded.

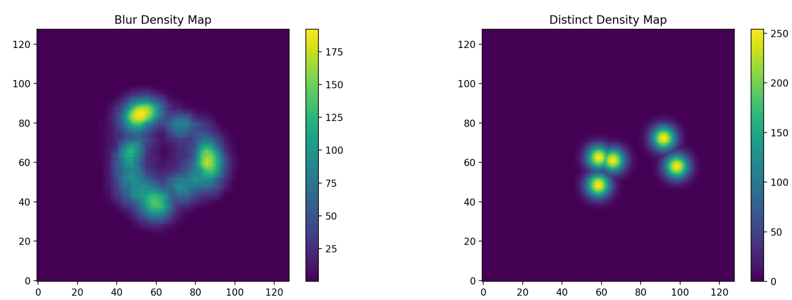
The three conditions above proves that the system is asymptotically stable under the PD controller $u_i = -k_p M_{V_i}(p_i - C_{V_i}) - k_d \dot{p}_i$

4. Implementation Details

In this section, we show the implementation of Lloyd gradient algorithm in our multi-robot coverage control problem, along with drawbacks of the simple Lloyd gradient algorithm and our proposal of heuristic approach to improve the robustness.

4.1. Density Map

Our simulation is performed on a set of density maps generated by a machine learning model.Two typical density maps are shown below:



(a) **(b)**

Fig. 1: Two representative density maps generated from machine learning model for coverage control simulation where (a) shows a blurred distribution scheme and (b) shows a distinct distribution scheme

4.2. Determination of Robot Count

We determine the number of agents to deploy by utilizing an adaptive threshold scheme to count the number of peaks in the density map.

4.3. Lloyd Algorithm

In this subsection we show the implementation of Lloyd algorithm to achieve the multi-robot coverage control. We first initialize n robots at random positions on the map, where n is the proper number of robots calculated in the previous subsection. For each robot, we further generate its Voronoi tessellation and compute its centroid. According to the Lloyd algorithm, we set the centroid as the target position and apply a PD controller drive robot towards it. The algorithm terminates if all robots stop moving or upon timeouts.Based on Lyapunov function from previous section, the PD controller will bring about local asymptotic stability.

Algorithm 1 Multi-Robot Coverage Control

```
for i ∈ {1, 2, ··· , N} do
  Initialize i^th agent's position p_i randomly on the map
end for
terminal ← False, success ← False, timestep ← 0
while not terminal do
  for i ∈ {1, 2, ··· , N} do
    if timestep ≡ 0 (mod 5) then
      Generate i^th agent's Voronoi tessellation, compute centroid C_{V_i} and mass M_{V_i}
      Target position t_i ← C_{V_i}
    end if
    Set control input u_i ← −k_p M_{V_i}(p_i − t_i) − k_d ṗ_i
    Update i^th agent's position p_i with u_i
  end for
  if u_i < eps for i ∈ {1, 2, ··· , N} then
    terminal ← True, success ← True, timestep ← timestep + 1
  else if timestep ⩾ MAX_TIMESTEP then
    terminal ← True, success ← False, timestep ← timestep + 1
  end if
end while
```

4.4. Exponential Fill

After applying Algorithm 1, we noticed a few cases where one robot get stuck at the corner for the map, and reaching steady state as the robot coincide with it's Voronoi centroid. Such scenarios happen because the robot's Voronoi tessellation cell lands completely in area of zero density, with $\rho(q) = 0, q \in V$. To circumvent the absence of density detection, we propose a modified density map with ubiquitous density distribution and negligible changes so that the Voronoi centroid is deviating towards the centroid of the global map if initialized in otherwise zero density area.

Algorithm 2 Exponential Fill of Density Map

```
q_{c_V} ← Centroid of the Density Map
for q ∈ Density Map do
  ρ(q) = ρ(q) + k_1 exp(−k_2 ‖q − q_{c_V}‖^2)
end for
```

By filling the whole map with exponentially decaying term with regards to squared distance from global centroid, any cell with otherwise zero density would be informed about the direction of the global centroid. The exponential term is tuned by scaling parameters k_1 and k_2, which determines exponential term magnitude and decaying sensitivity, respectively. Algorithm 2 provides sufficient information with arbitrarily small k_1 because density distribution is scale-invariant in determining centroid, and once any otherwise non-zero density pixel is enclosed in a cell upon update, the non-zero pixel would take dominance on Voronoi centroid, thus the exponential fill will not affect the original density map.

4.5. Global Planner

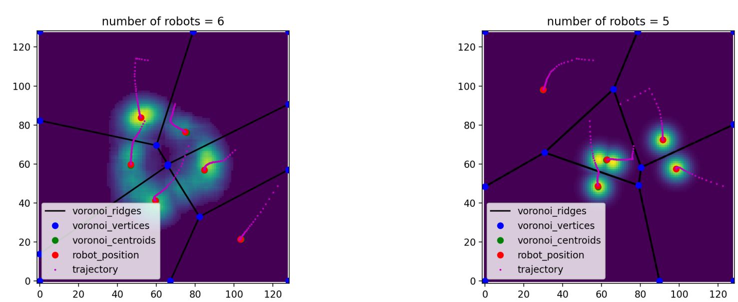
There are times when strictly decentralized control does not guarantee optimal result. A global planner that ensures robustness for systems with small amount of robots is introduced. Here we insert a step into Algorithm 1 which targets the robot pair with maximum and minimum cost and impose a pair of attractive force between them with magnitude proportional to their squared distance. The global planner solves stuck issues, speeds up initial iterations and stops when approaching steady state to avoid oscillations.

Algorithm 3 Global Controller

$flag \leftarrow True$
for $i \in \{1, 2, \cdots, N\}$ **do**
  if $timestep \equiv 0 \pmod 5$ **then**
    Generate i^{th} agent's Voronoi tessellation, compute centroid C_{V_i} and mass M_{V_i}
    Target position $t_i \leftarrow C_{V_i}$
  end if
  Set control input $u_i \leftarrow -k_p M_{V_i}(p_i - t_i) - k_d \dot{p}_i$
  if $flag$ **then**
    $i_{min} \leftarrow \arg\min_i \int_{V_i(p)} \|p_i - q\|^2 \phi(q)dq$, $i_{max} \leftarrow \arg\max_i \int_{V_i(p)} \|p_i - q\|^2 \phi(q)dq$
    Update control input $u_{i_{min}} \leftarrow u_{i_{min}} + k_{global}(p_{i_{max}} - p_{i_{min}}) \|p_{i_{max}} - p_{i_{min}}\|$
    Update control input $u_{i_{max}} \leftarrow u_{i_{max}} + k_{global}(p_{i_{min}} - p_{i_{max}}) \|p_{i_{max}} - p_{i_{min}}\|$
    if $\int_{V_{i_{min}}(p)} \|p_i - q\|^2 \phi(q)dq > k_{flag} \int_{V_{i_{max}}(p)} \|p_i - q\|^2 \phi(q)dq$ **then**
      $flag \leftarrow False$
    end if
  end if
  Update i^{th} agent's position p_i with u_i
end for

5. Simulations and Experiment Results

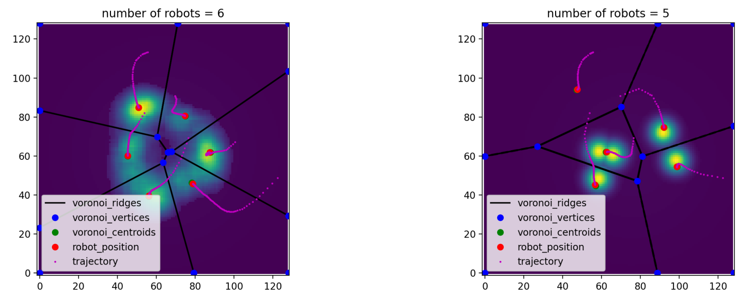
Several scenarios were simulated for each of the algorithms proposed above and their performances on the two density maps are displayed and analyzed below:



(a) **(b)**

Fig. 2: Robots get stuck in local optimal with Lloyd algorithm under both (a) blurred distribution scheme and (b) distinct distribution scheme

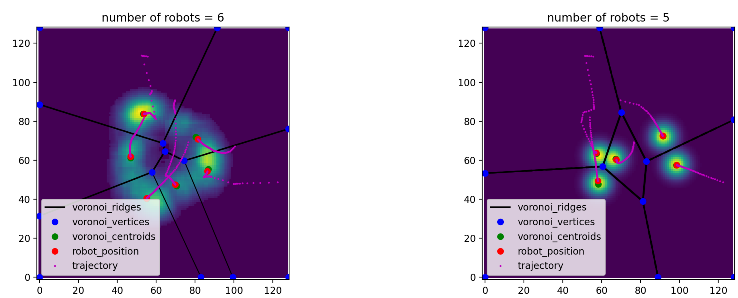
Above shows the performance of plain coverage control with Lloyd algorithms as in Algorithm 1. Both simulations show the issue of robots getting stuck at corners.



(a) **(b)**

Fig. 3: Density map with exponential fill brings about global optimal under (a) blurred distribution scheme but still fails in solving stuck problem under (b) distinct distribution scheme

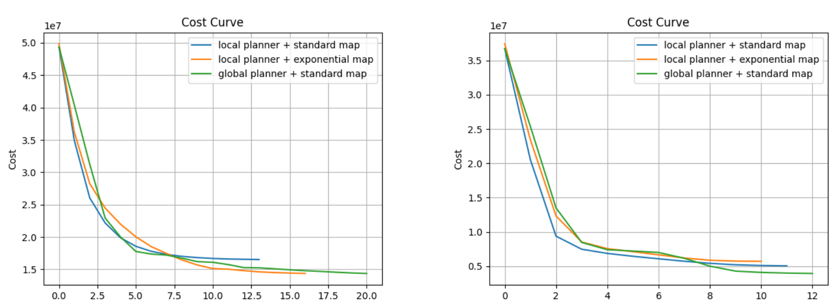
With exponential fill on density map, the robot in blurred map ceases to wander and reaches the optimal location; The impact of the exponential term on the lost robot in distinct map is not large enough to nudge the cell further, yet still provides a correct direction for the first few iterations.



(a) **(b)**

Fig. 4: Global planner enforces all robots to center under both (a) blurred distribution scheme and (b) distinct distribution scheme

After applying the global controller, both simulations display robust performances. The trajectories indicates the strong effect of global controller on correcting positions: the robots' final distribution and sequence on blurred map differ from that of the previous simulation with exponential fill.



(a) **(b)**

Fig. 5: Cost curves of three approaches under (a) blurred distribution scheme and (b) distinct distribution scheme, demonstrating strong performance improvement of exponential fill and global planner

The cost curves for both maps above quantify the optimality of each algorithms. All algorithms display a converge and descending cost, but exponential map and local planner both decrease cost if no stuck issue emerges. Global controller ensures lowest cost and rigid robustness but takes slightly longer time to reach steady state.

6. Conclusion

This paper proposes a collective control policy for groups of agents. Both exponential fill on density map and local planner improve robot performances, enhance robustness, and minimize global cost. The result of our research will assist the grander project and future study may employ extensive optimizations and especially achieve strictly decentralized and experimentally robust control policy. Working prototypes with specific functions could be tested as subsequent research.

References

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” IEEE Transactions on robotics and Automation, vol. 20, no. 2, pp. 243–255, 2004.

[2] Q.Du,V.Faber,andM.Gunzburger,“Centroidal voronoi tessellations: Applications and algorithms,” SIAM Rev., vol. 41, no. 4, pp. 637–676, Dec. 1999

[3] M. Santos, S. Mayya, G. Notomista and M. Egerstedt, ”Decentralized Minimum-Energy Coverage Control for Time-Varying Density Functions,” 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), New Brunswick, NJ, USA, 2019, pp. 155-161.