

Multi-Agent Ergodic Coverage in Urban Environments

Shivang Patel¹, Senthil Hariharan¹, Pranav Dhulipala¹
Ming C Lin^{1,2}, Dinesh Manocha^{1,2,3}, Huan Xu^{1,4,5} and Michael Otte^{1,2,4}

Abstract—An important aspect of dynamic urban coverage is how building collision avoidance is incorporated into the overall coverage mission. We consider a multi-agent urban dynamic coverage problem in which a team of flying agents uses downward facing cameras to observe the street-level environment outside of buildings. Cameras are assumed to be ineffective above a maximum altitude (lower than building height), such that agents must move around or over buildings to complete their mission. The main objective of this paper is to compare three different building avoidance strategies that are compatible with dynamic ergodic methods. To provide context for these results, we also compare our results to three other common coverage methods including: boustrophedon coverage (lawn-mower sweep), Voronoi region based coverage, and a naive grid method. All algorithms are evaluated in simulation with respect to four performance metrics (percent coverage, revisit count, revisit time, and the integral of area viewed over time), across team sizes ranging from 1 to 25 agents, and in five types of urban environments of varying density and height. We find that the relative performance of algorithms changes based on the ratio of team size to search area, as well the height and density characteristics of the urban environment.

Index Terms—Multi-Agent, Coverage, Lawn-mower, Ergodic, Boustrophedon, Voronoi, Urban Environment

I. INTRODUCTION

Multi-agent dynamic coverage in urban environments is used for missions such as: collecting scientific data, surface monitoring, and patrolling an area for intruders. *Ergodic* coverage is used for environmental monitoring and patrol scenarios in which we wish to obfuscate the revisit time to any particular point, for example, from an adversary. Buildings in urban environments present navigational challenges to performing ground-level ergodic coverage with small UAVs.

In this paper we consider a multi-agent ergodic coverage problem in which a team of identical UAV agents is tasked with visually monitoring an urban environment. In particular, outdoor street-level terrain. Each agent is equipped with a downward facing camera. Increasing camera elevation increases the size of the sensor’s ground-level footprint—increasing the area that can be monitored by a single agent but also causing a loss of image fidelity (as larger patches of ground-level terrain map to a single pixel). We are interested in the case where cameras are only useful up to a maximum elevation that is lower than the urban environment’s building

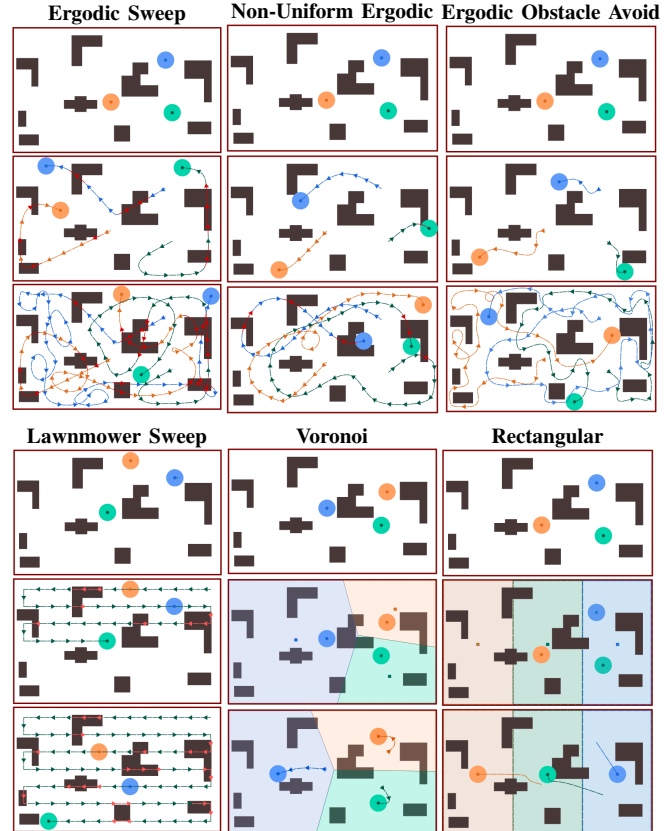


Fig. 1. **Multi-Agent Coverage Algorithms Evaluated:** three ergodic methods (top) and comparison algorithms (bottom). Top, middle, and bottom panels of each algorithms show agent movement at the start, middle, and end of a mission, respectively. Trajectories that fly over buildings are red. In the Voronoi and Rectangular methods, the regions are colored based on their corresponding robot. Ergodic methods are useful for adversarial scenarios because they obfuscate the revisit time to any particular point.

height. In other words, cases where agents must navigate over or around buildings during the mission.

The main contribution of this paper is the experimental comparison of six coverage methods in various urban environments (see Figure 1). We evaluate three multi-agent ergodic coverage methods that differ in their approach to obstacle avoidance, as well as lawn-mower sweep, Voronoi-based coverage, and a naive grid-based method. All six methods are experimentally evaluated in simulation. We analyze performance using four proposed metrics, across team sizes ranging from 1 to 25 agents, over five types of urban environments of varying density and height. We find that the relative performance of algorithms changes based on the ratio of team size to search area, as well the height and

Authors are with: ¹Maryland Robotics Center, ²The Department of Computer Science, ³The Department of Electrical and Computer Engineering, ⁴The Department of Aerospace Engineering, and ⁵The Institute for Systems Research, University of Maryland, College Park. spatel43@umd.edu, {lin, dm}@cs.umd.edu, {mumu, otte}@umd.edu. This work was supported by DARPA cooperative agreement HR00111820028 as part of DARPA OFFSET, ARO grant W911NF-19-1-0069.

density characteristics of the urban environment.

The rest of this paper is organized as follows: Section II reviews related work. Preliminaries, appear in Section III. The algorithms evaluated are described in Section IV. Experiments are presented in section V, and a discussion of results appears in Section V-B.

II. RELATED WORK

We assume a broad definition of the term ‘coverage,’ and use it to indicate the set of problems including *coverage*, *surveillance*, and *environmental monitoring*. In *dynamic coverage*, agent(s) move continuously throughout the environment so that every point in the environment is intermittently observed [1]–[5]. In contrast, in *static coverage* agent(s) move to a set of advantageous locations, and then remain at the same locations for the duration of the mission [6]–[9].

Ergodic coverage is applicable to adversarial coverage problems, such as patrol [4], and uses control laws to balance search efficiency and unpredictability (since, in an adversarial scenario, the practice of revisiting points on a set schedule can be exploited by the adversary). Previous work on ergodic coverage has considered search and target localization related problems [10]–[14] but not problems involving ground-level coverage tasks in 3D urban environments.

Ergodic sweep control laws are often designed as a function of a target coverage distribution and the current coverage distribution. Biasing coverage to obstacle-free [4] reduces obstacle collisions, but does not eliminate collisions. A method for explicit obstacle avoidance within ergodic coverage is presented in [5], which combines ergodic coverage control laws with phisiocomemetic vector fields to “repel” agents away from obstacles. Three of the six algorithms that we compare are ergodic coverage variants that, respectively: (1) ignore obstacles during coverage planning and then fly up and over them, (2) bias search away from obstacles and then fly up and over them if necessary, (3) avoid obstacles by flying around them without a change in elevation.

Lawnmower sweep (or *boustrophedon coverage*) uses a simple back and forth motion. Assuming a sweep sensor has positive radius, this can be used to cover any obstacle-free convex environment of finite area [1]. Polygonal obstacles can be addressed by partitioning the free space into a finite number of convex regions, and sweeping each region separately [2]. In [3] a multi-agent team is divided into two groups of agents, one performing exploration and the other coverage. Most lawnmower sweep implementations assume a 2D environment such that regions surrounded by obstacles are topologically separated from the rest of the environment.

Other methods of generating a coverage trajectory include: Zelinski’s algorithm, which orders the sweep based on the levels sets of a wave-front expansion [15], and ideas motivated by space filling curves [16].

Static coverage algorithms partition the search space into regions such that there is a mapping from robots to regions, and each robot can view its region from a stationary position.

Centralized methods [6], [7], [17] compute the search space division on a single agent or server, and then send the

solution to all agents at runtime or *a priori*. A cellular decomposition of the environment followed by the calculation of a multi-robot spanning forest is used by [7], and each robot is assigned one tree in the forest. A segmentation technique is used in [17], where each robot is assigned a unique segment. Work in [18] partitions the environment using the weighted K-Means clustering algorithm. Centralization has the disadvantage of introducing single points of failure.

In contrast, decentralized and/or distributed process do not have a single point-of-failure. In many strategies robots start at their initial locations and then tend toward a multi-robot configuration with desirable coverage properties over time. Iterative Voronoi-based approaches achieve such a distributed control strategy [19]–[31]. One advantage of such methods is that each agent need only communicate with its Voronoi neighbors. The basic Voronoi method is one of the six methods we compare in this paper.

Information gathering [32], [33] is closely related to coverage, the main difference being that the amount of information that can be gathered at a particular location is non-uniform and changes as a function of each sensor measurement.

In target search [34]–[37] the goal is to locate one or more stationary or moving targets. Target search problem variants that share similarities with the coverage problem include probability based [38], [39], information based [40]–[42], and game theoretic formulations [43].

Exploration is similar to coverage in that both problems are concerned with visiting all points in the environment [44]. However, coverage problems often require repeated visits to each location, while exploration is completed once each location has been visited once.

In previous work [45] we present a collision avoidance algorithm for a swarm of UAVs performing an urban coverage task. The method in [45] focuses on agent-to-agent and agent-to-obstacle local collision avoidance. The method in [45] can be used as a post-processing step for all of the algorithms considered in the current paper.

III. NOMENCLATURE AND PROBLEM STATEMENT

A team of n robots is the set $R = \{r_1, \dots, r_n\}$, where r_i denotes the i -th robot. The 3D workspace is χ^3 and contains the mutually disjoint obstacle space $\chi_{obs}^3 \subset \chi^3$ and the free space $\chi_{free}^3 \subset \chi^3$, i.e., $\chi_{obs}^3 \cap \chi_{free}^3 = \emptyset$ and $\chi_{obs}^3 \cup \chi_{free}^3 = \chi^3$. We assume the ground plane of the workspace can be approximated by 2D Euclidean space \mathbb{R}^2 . Let the projections of χ^3 and χ_{obs}^3 directly down onto \mathbb{R}^2 be denoted χ and χ_{obs} , respectively. The free space at ground level is defined $\chi_{free} = \chi \setminus \chi_{obs}$ such that $\chi_{obs} \cap \chi_{free} = \emptyset$ and $\chi_{obs} \cup \chi_{free} = \chi$. We assume an urban environment such that χ_{obs}^3 contains buildings. The area of desired coverage is defined as all ground-level terrain (and not the sides or tops of buildings), and denoted $\chi_{search} \equiv \chi_{free}$.

Each robot r_i is assumed to be a point, and follow a trajectory in χ_{free}^3 while using a downward facing camera sensors to observe χ_{free} . We assume a continuous time model starting at $t = 0$. The interval of time from the beginning of the mission until $t = t_{max}$ is $[0, t_{max}]$. Trajectory ρ_i

denotes the geometric set of points along the curve in χ_{free}^3 that is traced out by robot r_i over time, such that $\rho_i \subset \chi_{free}^3$. The multipath ψ is the set of all robots' trajectories. Geometrically $\psi = \bigcup_{i=1}^n \{r_i\}$. The space containing all multipaths is $\Psi = \bigcup \{\psi\}$.

We assume robots have identical downward facing camera sensors such that when r_i flies at altitude h the projection of r_i 's field-of-view down onto χ_{search} is a disc B_h of radius $f(h)$. We assume sensor will be used at the highest altitude \tilde{h} for which it remains a reliable sensor (increasing altitude increases B_h , which is advantageous, but may degrade sensor reliability, which is disadvantageous). We drop the subscript when the sensor is used at the optimal altitude, $B = B_{\tilde{h}}$. Agents may increase their altitude $h > \tilde{h}$ to fly over obstacles, but cannot observe χ_{free} while at altitude $h > \tilde{h}$. Let $\tilde{\rho}_i \subset \rho_i$ be the subset of ρ_i containing all points at which robot r_i 's camera sensor is functional, i.e., for which r_i is at altitude $h = \tilde{h}$. Similarly, let $\tilde{\psi} = \bigcup_{i=1}^n \{\tilde{\rho}_i\}$. The area swept by the team is then:

$$\chi_{swept} = (\tilde{\psi} \oplus B) \cap \chi_{search} \quad (1)$$

where ' \oplus ' denotes the Minkowski sum¹. The intersection with χ_{search} is included in Equation 1 so that $\chi_{swept} \subset \chi_{search}$ by construction.

Let $\tilde{\Psi} = \bigcup \{\tilde{\psi}\}$ be the space containing all $\tilde{\psi}$. Let $\mathbf{1}_{see}$ be an indicator function that returns 1 or 0 based on whether or not, at time t , a point $x \in \chi_{search}$ is observed by at least one robot in the team. $\mathbf{1}_{see} : \chi_{search} \times [0, t_{max}] \times \tilde{\Psi} \rightarrow \{1, 0\}$, and so $\mathbf{1}_{see}(x, t, \tilde{\psi}) \mapsto 1 \iff x \in \psi(t) \oplus B$ and $\mathbf{1}_{see}(x, t, \tilde{\psi}) \mapsto 0 \iff x \notin \psi(t) \oplus B$.

Given a particular $\tilde{\psi}$, the function $g_{\tilde{\psi}, t_{max}}$ is a map from the search space χ_{search} to the time-duration domain, $g_{\tilde{\psi}, t_{max}} : \chi_{search} \rightarrow [0, t_{max}]$ that measures the cumulative time that point $x \in \chi_{search}$ is observed by at least one robot, $g_{\tilde{\psi}, t_{max}}(x) = \int_0^{t_{max}} \mathbf{1}_{see}(x, t, \tilde{\psi}) dt$, where the integral is Lebesgue. Dynamic coverage algorithms can be characterized by a requirement that points in χ_{search} be visited infinity often as $t_{max} \rightarrow \infty$. Given continuous vehicles trajectories, this is $\lim_{t_{max} \rightarrow \infty} g_{\tilde{\psi}, t_{max}}(x) = \infty$ for all $x \in \chi_{search}$.

The dynamic coverage problem is formally defined below.

Problem 1. The *dynamic multi-robot urban coverage problem*: Given a team of n robots $R = \{r_1, \dots, r_n\}$, an urban environment χ^3 with obstacle space and free space such that $\chi_{obs}^3 \cap \chi_{free}^3 = \emptyset$ and $\chi_{obs}^3 \cup \chi_{free}^3 = \chi^3$, and with ground-level search space $\chi_{search} = \chi_{free}$, determine a multipath $\psi = \{r_1, \dots, r_n\}$ such that $\chi_{swept} = (\tilde{\psi} \oplus B) \cap \chi_{search}$ and where, for all $x \in \chi_{search}$, $\lim_{t_{max} \rightarrow \infty} g_{\tilde{\psi}, t_{max}}(x) = \infty$.

IV. ALGORITHMS

The three ergodic methods we evaluate are described in Section IV-A. The algorithms we use for comparison appear in Sections IV-B, IV-C, and IV-D, respectively.

¹The Minkowski sum of two geometric objects is found by dilating (increasing the size) of one object by that of the other. It is formally defined as the set containing all points described by position vectors $a + b$ where a is a position vector describing a point in the first object and b is a position vector describing a point in the second object.

A. Multi-Agent Urban Ergodic Coverage

In ergodic coverage the agent/team follows trajectories such that the relative time spent in each non-zero measure region of the environment can be prescribed by a user. The desired properties only hold almost surely in the limit as time approaches infinity. Thus, practical performance can be expected to improve with mission duration.

Ergodic coverage is most easily described—and implemented—as an evolutionary process that generates a trajectory. The subroutine `singleErgodic(f_χ)`, described in Algorithm 1, computes the ergodic trajectory for a single agent, assuming a user defined coverage distribution $f_\chi(x)$ over χ . The ' \circ ' symbol denotes trajectory concatenation.

The subroutine `nextStep(f_χ, M_k, C_k)` (on line 4 of Algorithm 1) implements the first order control laws of ergodic coverage [4] for a single agent without explicit obstacle avoidance. This is calculated $B_j(t) = \sum_R \Lambda_k S_k \nabla f_k(x_j(t))$ which is further normalized and constrained by velocity u_{max} $u_j(t) = -u_{max} \frac{B_j(t)}{\|B_j(t)\|_2}$, Where Λ_k is constant and $S_k(t)$ is difference between the current distribution and target distribution, given by $S_k(t) := C_k(t) - M_k(t)$. $\nabla f_k(x_j(t))$ is gradient of the Fourier basis function which is given by

$$\nabla f_k(x_j(t)) = \frac{1}{h_k} \begin{bmatrix} -k_1 \sin(k_1 x_1) \cos(k_2 x_2) \\ -k_2 \cos(k_1 x_1) \sin(k_2 x_2) \end{bmatrix}$$

and the Fourier basis function is given by $f_k(x) = \frac{1}{h_k} \cos(k_1 x_1) \cos(k_2 x_2)$. The Fourier coefficient $C_k(t)$ is calculated $C_k(t) = \sum_{j=1}^N \int_0^t f_k(x_j(\tau)) d\tau$ and Fourier coefficient of target distribution is $M_k(t) := N t \mu_k$, where $k_1 = \frac{K_1 \pi}{L_1}$, $k_2 = \frac{K_2 \pi}{L_2}$, and $\mu_k = \langle \mu, f_k \rangle$, where $\langle \cdot, \cdot \rangle$ is an inner product. $h_k = \left(\int_0^{L_1} \int_0^{L_2} \cos^2(k_1 x_1) \cos^2(k_2 x_2) \right)^{1/2}$.

The subroutine `singleErgAvoidObs(χ, χ_{obs})`, described in Algorithm 2, computes an ergodic trajectory for a single agent that explicitly avoids obstacles by using an obstacle repulsive feedback law which has derived by [5] (line 5), and calculated as follow: This feedback law implementing obstacle avoidance would be governed by $V_j^*(t) := -\alpha V_j(t) + (1 - \alpha) F_j^o(r_j)$ where $V_j(t) := -\frac{B_j(t)}{\|B_j(t)\|_2}$ and $F_j^o(r_j)$ is a repulsive vector field which is defined as,

$$F_j^o(r_j) = \begin{cases} F|_{\lambda=1}(r_j - r_o) & \text{if } P^T(r_j - r_o) \geq 0 \\ F|_{\lambda=0}(r_j - r_o) & \text{if } P^T(r_j - r_o) < 0 \end{cases}$$

where function $F(r) = \lambda(P^T r)r - p(r^T r)$ as defined in [46]. Here $\lambda \in R$ specifies shape of the vector field, $r = r_j - r_o$ which is distance between robot and obstacle, and $P \in R^2$. According to [5], this method will only works for a defined shaped object.

In this paper, we extend this idea of defined shaped obstacle avoidance to accommodate polygonal obstacles. A particular obstacle O_{arb} is defined by a combination of N shaped obstacles: $O_{arb} = \bigcup_{i=1}^N \{o_i\}$ for some positive integer N . The feedback law of a single obstacle to multiple

Algorithm 1 singleErgodic(f_χ) 1: $M_k, C_k \leftarrow \text{initErgParams}(f_\chi)$ 2: $\rho \leftarrow \emptyset$ 3: for $t = 1, \dots, t_{\max}$ do 4: $\rho_{[0,t]} \leftarrow \rho_{[0,t-1]} \circ \text{nextStep}(f_\chi, M_k, C_k)$ 5: $C_k \leftarrow \text{updateCurrentDists}(C_k, \rho_{[t]})$	Algorithm 4 Biased Multi-Agent Ergodic 1: for $i \leftarrow 1 \dots n$ do 2: $\rho_i \leftarrow \text{singleErgodic}(f_{\chi_{free}})$ 3: $\tilde{\rho}_i \leftarrow \text{flyOverBldgs}(\rho_i, \chi_{free}^3, \chi_{obs}^3)$ 4: $\tilde{\psi} \leftarrow (\tilde{\rho}_1, \dots, \tilde{\rho}_n)$	Algorithm 7 Multi-Agent Voronoi Cover 1: $\mathbf{x}_0 \leftarrow \text{projectOnto}(R, \chi)$ 2: $\{\rho_0, \dots, \rho_n\} \leftarrow \{x_{0,1}, \dots, x_{0,n}\} \leftarrow \mathbf{x}_0$ 3: for $t = 1, \dots, t_{\max}$ do 4: $\{C_1, \dots, C_n\} \leftarrow \text{voronoiPart}(\mathbf{x}_{t-1}, \chi)$ 5: for $i \leftarrow 1, \dots, n$ do 6: $x_{t,i} \leftarrow \text{centroid}(C_i)$ 7: $\rho_i = \rho_i \circ x_{t,i}$ 8: $\mathbf{x}_t \leftarrow \{x_{t,1}, \dots, x_{t,n}\}$ 9: for $i \leftarrow 1, \dots, n$ do 10: $\tilde{\rho}_i \leftarrow \text{flyOverBldgs}(\rho_i, \chi_{free}^3, \chi_{obs}^3)$ 11: $\tilde{\psi} \leftarrow (\tilde{\rho}_0, \dots, \tilde{\rho}_n)$
Algorithm 2 singleErgAvoidObs($f_{\chi_{free}}, \chi_{obs}$) 1: $M_k, C_k \leftarrow \text{initErgParams}(f_{\chi_{free}})$ 2: $\rho_{[0,t]} \leftarrow \emptyset$ 3: for $t = 1, \dots, t_{\max}$ do 4: $\rho_{[t-1,t]} \leftarrow \text{nextStep}(f_{\chi_{free}}, M_k, C_k)$ 5: $\rho_{[0,t]} \leftarrow \text{calcVectorField}(\rho_{[t-1,t]}, \chi_{obs})$ 6: $C_k \leftarrow \text{updateCurrentDists}(C_k, \rho_{[t]})$	Algorithm 5 Obstacle Avoiding Multi-Agent Ergodic 1: for $i \leftarrow 1 \dots n$ do 2: $\rho_i \leftarrow \text{singleErgAvoidObs}(f_{\chi_{free}}, \chi_{obs})$ 3: $\tilde{\psi} \leftarrow \psi \leftarrow (\rho_1, \dots, \rho_n)$	Algorithm 8 Multi-Agent Grid Cover 1: $\mathbf{x}_0 \leftarrow \text{projectOnto}(R, \chi)$ 2: $\{\rho_0, \dots, \rho_n\} \leftarrow \{x_{0,1}, \dots, x_{0,n}\} \leftarrow \mathbf{x}_0$ 3: $\{C_1, \dots, C_n\} \leftarrow \text{gridPartition}(\mathbf{x}_0, \chi)$ 4: for $i \leftarrow 1, \dots, n$ do 5: $x_{1,i} \leftarrow \text{centroid}(C_i)$ 6: $\rho_i = \rho_i \circ x_{1,i}$ 7: $\tilde{\rho}_i \leftarrow \text{flyOverBldgs}(\rho_i, \chi_{free}^3, \chi_{obs}^3)$ 8: $\tilde{\psi} \leftarrow (\tilde{\rho}_0, \dots, \tilde{\rho}_n)$
Algorithm 3 Multi-Agent Ergodic 1: for $i \leftarrow 1 \dots n$ do 2: $\rho_i \leftarrow \text{singleErgodic}(f_{\chi_{vacant}})$ 3: $\tilde{\rho}_i \leftarrow \text{flyOverBldgs}(\rho_i, \chi_{free}^3, \chi_{obs}^3)$ 4: $\tilde{\psi} \leftarrow (\tilde{\rho}_1, \dots, \tilde{\rho}_n)$	Algorithm 6 Multi-Agent Lawnmower 1: $\rho \leftarrow \text{Boustrophedon}(\chi_{vacant})$ 2: $\tilde{\rho} \leftarrow \text{flyOverBldgs}(\rho, \chi_{free}^3, \chi_{obs}^3)$ 3: $\ell \leftarrow \ \tilde{\rho}_1\ $ 4: for $i \leftarrow 1 \dots n$ do 5: $\tilde{\rho}_i \leftarrow \text{rotateCycle}(\tilde{\rho}, \frac{i}{n}\ell)$ 6: $\tilde{\psi} \leftarrow (\tilde{\rho}_1, \dots, \tilde{\rho}_n)$	

obstacles is then given by

$$V_j^*(t) := \left(\prod_{i=1}^N \alpha^{o_i} \right) V_j(t) + \sum_{i=1}^N (1 - \alpha^{o_i}) F_j^{o_i}(r_j)$$

The obstacle avoidance control law is then normalized and constrained by velocity u_{max} as $u_j(t) = -u_{max} \frac{V_j(t)}{\|V_j(t)\|_2}$. The parameter $\alpha \in [0, 1]$ is a bump factor taking value 1 if the robot is far from the obstacle and reducing to 0 when approaching an obstacle (in the vector field). While we have chosen to use a linear α in this paper, it is also possible to define α in other ways [5].

Let the probability distribution functions $f_{\chi_{vacant}}(x)$ and $f_{\chi_{free}}(x)$ respectively define uniform random distribution over the entire space (ignoring obstacles) and over the free space (biasing movement away from obstacles).

$$f_{\chi_{vacant}}(x) = \begin{cases} \frac{1}{\|\chi\|} & \text{if } x \in \chi \\ 0 & \text{if } x \notin \chi \end{cases}$$

$$f_{\chi_{free}}(x) = \begin{cases} \frac{1}{\|\chi_{free}\|} & \text{if } x \in \chi_{free} \\ 0 & \text{if } x \notin \chi_{free} \end{cases}$$

We compare three versions of multi-agent ergodic sweep for that differ based on how urban obstacles are handled:

- 1) **Multi-Agent Urban Ergodic Sweep** is a simple repairing strategy that calculates a trajectory assuming no obstacles exists (Algorithm 3). This method modifies its solution to avoid obstacles in χ^3 using the subroutine $\text{flyOverBldgs}(\rho, \chi_{free}^3, \chi_{obs}^3)$ on line 3. The effect of this subroutine is to generate a building avoiding trajectory $\tilde{\rho}$ by replace each obstacle intersecting segment of the original trajectory ρ with a different segment along which the agent increase its elevation above the building, flies over it, and then descends back to the optimal sweep elevation.

- 2) **Multi-Agent Urban Biased Ergodic Sweep** is similar, except that we define the area of desired uniform coverage to be χ_{free} instead of χ_{vacant} (Algorithm 4, line 2). This biases ρ_i away from obstacles, but does not eliminate the need to occasionally fly up and over a buildings. $\tilde{\rho}_i$ is created from ρ_i by having agent i fly over buildings when necessary (Algorithm 4, line 3).
- 3) **Multi-Agent Obstacle Avoiding Urban Ergodic Sweep** forces each ρ_i to avoid obstacles using the vector field algorithm (Algorithm 5). In other words, by using $\text{singleErgAvoidObs}(f_{\chi_{free}}, \chi_{obs})$ to prevent ρ_i from intersecting χ_{obs} . In our experiments, all agents fly at the optimal sweep elevation such that isolated internal courtyards are not visited.

B. Multi-Agent Urban Lawnmower Coverage

Single agent lawnmower sweep in a 2-dimensional environment is known as boustrophedon coverage. Our multi-agent implementation is presented in Algorithm 6, and uses the single agent boustrophedon algorithm as a subroutine. We start by finding a single agent boustrophedon coverage cycle of an obstacle free environment χ_{vacant} , where χ_{vacant} has the same footprint as χ (line 1). $\tilde{\rho}_i$ is created from ρ_i by having agent i rise in elevation to fly over a building and descend back to the sweep altitude afterward (line 2). Given a single agent cycle $\tilde{\rho}_1$ of length ℓ , the cycles of the other agents are calculated by rotating the starting positions $\frac{i}{n}$ of the length around the cycle for agent i (lines 3-5). The obstacle avoiding multi-path of the team is given by the n -tupal containing the single agent trajectories (line 6).

C. Voronoi Urban Coverage Algorithm

Given a set of n generating points (we use the set $\mathbf{x} = \{x_1, \dots, x_n\}$ of robot projections onto χ) a Voronoi space partitioning of χ creates mutually disjoint open cells by assigning each point $x \in \chi$ to a cell C_i associated

TABLE I
ALGORITHMS

	Name of Algorithm
1	Lawnmower
2	Ergodic
3	Biased Ergodic
4	Obstacle Avoiding Ergodic
5	Voronoi
6	Rectangular

TABLE II
EVALUATION METRICS

	Name of Metric
1	Coverage Area (percentage of total area)
2	Number of Visits
3	Mean Duration Between Visits to Each Points
4	Mean Time Spent Near Each Point

TABLE III
URBAN ENVIRONMENTS IN EXPERIMENTS

	Height	Density	Dimensions	Buildings
1	Tall	High	$50.96 \times 39.33 \times 29.50$	27
2	Tall	Low	$56.25 \times 53.03 \times 14.25$	16
3	Short	High	$64.26 \times 53.80 \times 12.50$	79
4	Short	Low	$96.67 \times 62.92 \times 7.2$	23
5	Mixed	Mixed	$147 \times 59 \times 44.6$	28

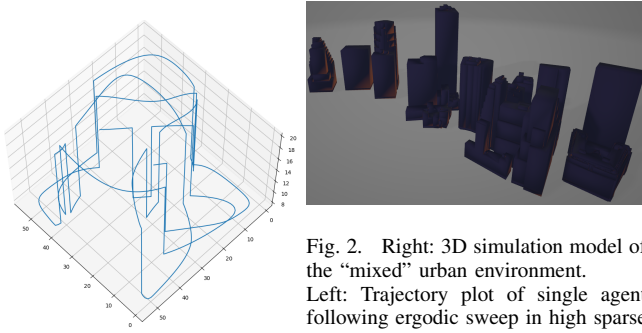


Fig. 2. Right: 3D simulation model of the “mixed” urban environment. Left: Trajectory plot of single agent following ergodic sweep in high sparse simulated urban environment.

with the closest generating point (Algorithm 7, line 4). $C_i = \{x \mid \|x - x_i\| < \|x - x_{j \neq i}\|\}$. The Voronoi partitioning of χ , i.e., instead of all of \mathbb{R}^2 , is found by placing reflected versions of points across each boundary of χ , and truncating the resulting extended Voronoi diagram to the footprint of χ . When χ is a rectangle this requires $5n$ points.

Voronoi coverage methods work by having each robot move a small distance toward the centroid of its current voronoi cell, recalculate new voronoi cells (Algorithm 7, lines 4-7), and then repeat (lines 3-8). Over time, this causes robots to greedily space themselves away from their neighbors.

D. Grid-Based Urban Coverage Algorithm

We also implement a naive grid-based algorithm for static coverage. This method uses rectangles, instead of Voronoi regions, to divide the space among agents (Algorithm 8). This rectangular division of the space can be calculated *a priori*. At runtime each agent moves to the center of its assigned rectangular region.

V. EXPERIMENTS

In this section we compare the six multi-agent coverage methods (see Tables I and II) across a variety of environments (see Table III), for teams of size 1 to 25 agents. Environments differ based on building height, building density, and footprint size. Each combination of Environment, Algorithm, and team size is repeated over three random trials.

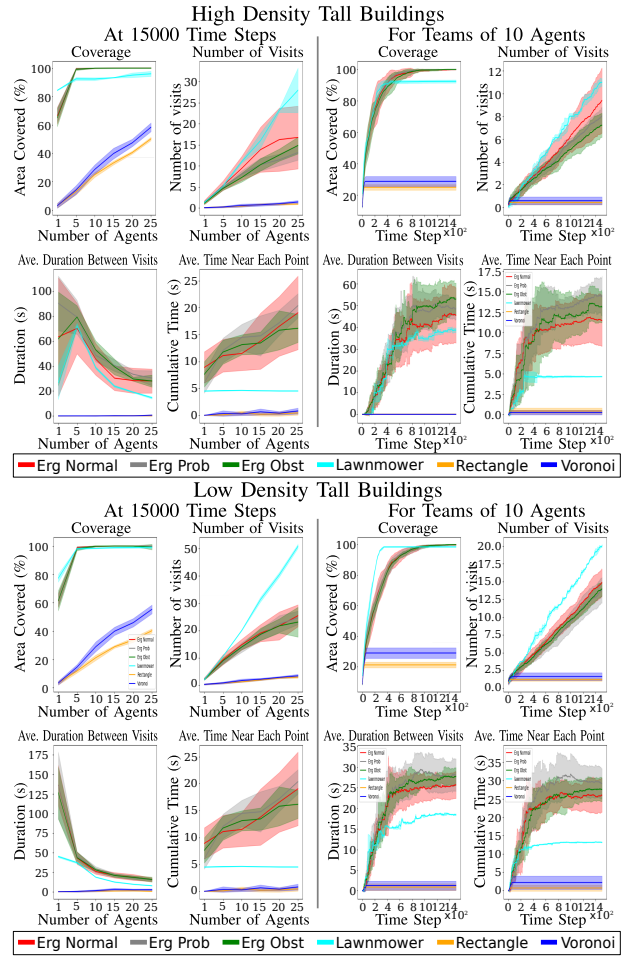


Fig. 3. Performance of different size teams after 15000 time steps (left) and performance for 10 agents over time (right) in environments with tall buildings and high building density (top 4 rows) or low building density (bottom 4 rows) density. In tall environments it is better to fly around buildings than over them. Greater building density amplifies performance differences between methods.

Experiments are run in the Ubuntu Linux operating system using Robot Operating System (ROS), Gazebo, and the PX4. UAVs measure 1 X 1 X 0.3 meters.

A. Performance Metrics

We empirically evaluate performance with respect to four proposed metrics, by randomly sampling a large number of points in the environment and tracking statistics in a disc surrounding each point. Statistical results for a particular trial are obtained by integrating over these points.

- 1) **Percent Coverage:** The percentage of the map (point regions) that has been swept at least once.
- 2) **Visits Count.** The total number of visits a point's region has been visited.
- 3) **Revisit Time.** The time duration between successive visits to a point's region.
- 4) **Time spent.** The cumulative time that any agent is within a point's region. We unpaue the counter when any agent enters point's region and pause the counter when the agent leaves that region.

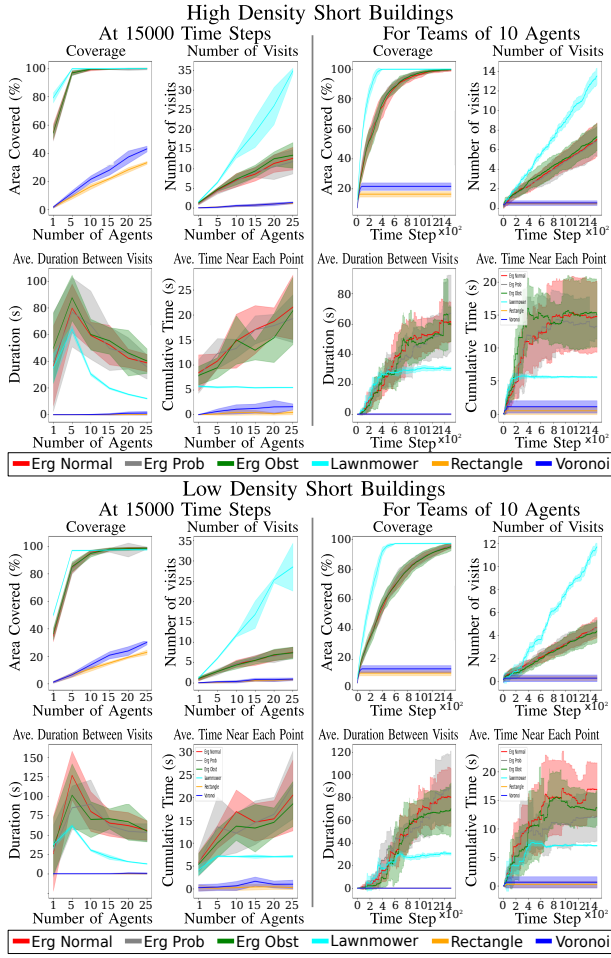


Fig. 4. Performance of different size teams after 15,000 time steps (left) and performance for 10 agents over time (right) in environments with short buildings and high building density (top 4 rows) or low building density (bottom 4 rows) density. Biased ergodic sweep outperforms the other ergodic methods in short environments.

Agent starting locations are chosen randomly for all methods except for Lawnmower Sweep. For Lawnmower Sweep, the initial coordinates of the agents are randomly chosen along the lawnmower trajectory (this causes the lawnmower algorithm to have a slight start-up advantage over other methods because it eliminates the startup phase in which agent travel to their equally spaced positions along the cycle). Experimental results appear in Figures 2-5. We discuss our main observations next.

B. Results

The key findings of our experiments are:

- In tall environments, flying around buildings is better than flying over them.
- In short environments, the best performing ergodic method is to bias movement toward the free space, while not explicitly avoiding obstacles—i.e., flying over buildings is allowed in the rare cases the ergodic path moves through the obstacles space.
- Building height effects become more pronounced as team size increases.

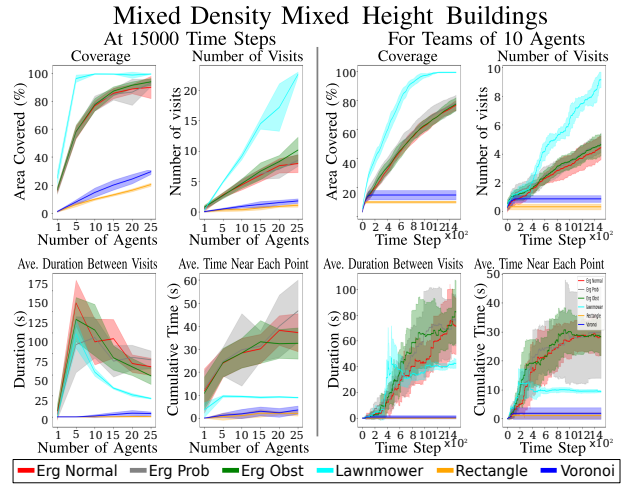


Fig. 5. Performance of different size teams (left) and performance over time (right) in environments with mixed density and mixed buildings.

- Increasing building density amplifies the differences between the different methods.
- For the dynamic methods, mean revisit times appear to scale inversely proportional to agent number.

C. The Performance Of Ergodic vs. Non-Ergodic Methods

The Lawnmower sweep, Voronoi, and Grid methods provide context to understand the performance differences between the various ergodic methods we compare, namely:

- Lawnmower sweep algorithm uses systematic but predictable motion, and provide an upper bound on the amount of area that can be covered.
- Voronoi and Grid methods are designed to solve static problems and provide a reasonable lower bound on what is achievable without movement.
- The ergodic methods revisit a particular point less often, on average, than lawnmower sweep, but more often than the Voronoi or grid methods.
- Ergodic methods are preferable to lawn-mower sweep in adversarial scenarios.

VI. SUMMARY AND CONCLUSION

In this paper, we empirically evaluate multi-agent ergodic algorithms for dynamic coverage in urban environments. We compare three ergodic methods that differ in their strategy of handling of building obstacle avoidance. In contrast to prior investigations on purely 2D areas or within a 3D volume, we consider a different scenario in which agents move in 3D but the coverage region is defined as the 2D ground plane.

We find that the performance of coverage algorithms in urban environments is affected by the strategy used to avoid building collisions. In environments with tall buildings it is better to fly around the buildings than over them. In environments with short buildings it is better to bias ergodic motion away from obstacles and then greedily fly over buildings when necessary. We also find that mean revisit times appear to scale inversely proportional to agent number.

REFERENCES

- [1] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [2] A. Ntawumenyikizaba, H. H. Viet, and T. Chung, "An online complete coverage algorithm for cleaning robots based on boustrophedon motions and a* search," in *2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012)*, vol. 2. IEEE, 2012, pp. 401–405.
- [3] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, 2008.
- [4] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4-5, pp. 432–442, 2011.
- [5] H. Salman, E. Ayvali, and H. Choset, "Multi-agent ergodic coverage with obstacle avoidance," in *Twenty-Seventh International Conference on Automated Planning and Scheduling*, 2017.
- [6] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 735–741.
- [7] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 77–98, 2001.
- [8] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards robust on-line multi-robot coverage," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1710–1715.
- [9] D. W. Gage, "Command control for many-robot systems," Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA, Tech. Rep., 1992.
- [10] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2015.
- [11] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon ergodic exploration," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 62–80, 2018.
- [12] S. Ivić, B. Crnković, and I. Mezić, "Ergodicity-based cooperative multiagent area coverage via a potential field," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1983–1993, 2017.
- [13] S. Ivić, A. Andrejčuk, and S. Družeta, "Autonomous control for multi-agent non-uniform spraying," *Applied Soft Computing*, vol. 80, pp. 742–760, 2019.
- [14] S. Ivić, "Motion control for autonomous heterogeneous multi-agent area search in uncertain conditions," *arXiv preprint arXiv:1911.09137*, 2019.
- [15] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proceedings of international conference on advanced robotics*, vol. 13, 1993, pp. 533–538.
- [16] S. V. Spire and S. Y. Goldsmith, "Exhaustive geographic search with mobile robots along space-filling curves," in *Collective robotics*. Springer, 1998, pp. 1–12.
- [17] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 1160–1165.
- [18] A. Solanas and M. A. Garcia, "Coordinated multi-robot exploration through unsupervised clustering of unknown space," in *IROS*. Cite-seer, 2004, pp. 717–721.
- [19] M. Schwager, J.-J. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3289–3294.
- [20] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [21] A. Okabe and A. Suzuki, "Locational optimization problems solved through voronoi diagrams," *European Journal of Operational Research*, vol. 98, no. 3, pp. 445–456, 1997.
- [22] 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.
- [23] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM review*, vol. 41, no. 4, pp. 637–676, 1999.
- [24] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *robotics: science and systems*, 2006, pp. 49–56.
- [25] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 3947–3952.
- [26] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 61–75.
- [27] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4982–4989.
- [28] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [29] I. Kamon, E. Rimon, and E. Rivlin, "Tangentbug: A range-sensor-based navigation algorithm," *The International Journal of Robotics Research*, vol. 17, no. 9, pp. 934–953, 1998.
- [30] K. Guruprasad, Z. Wilson, and P. Dasgupta, "Complete coverage of an initially unknown environment by multiple robots using voronoi partition," in *International Conference on Advances in Control and Optimization in Dynamical Systems*, 2012.
- [31] K. Guruprasad and D. Ghose, "Automated multi-agent search using centroidal voronoi configuration," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 420–423, 2011.
- [32] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.
- [33] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *Control of Network Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 335–347, 2015.
- [34] B. Koopman, "The theory of search. ii. target detection," *Operations research*, vol. 4, no. 5, pp. 503–531, 1956.
- [35] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, 2011.
- [36] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *Emerging Security Technologies (EST), 2010 International Conference on*, Sept 2010, pp. 142–147.
- [37] S. Nayak, S. Yeotikar, E. Carrillo, E. Rudnick-Cohen, M. K. M. Jaffar, R. Patel, S. Azarm, J. Herrmann, H. Xu, and M. W. Otte, "Experimental comparison of decentralized task allocation algorithms under imperfect communication," *IEEE Robotics and Automation Letters*, vol. to appear, 2020.
- [38] T. H. Chung and J. W. Burdick, "Analysis of search decision making using probabilistic search strategies," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 132–144, Feb 2012.
- [39] P. Dames and V. Kumar, "Autonomous localization of an unknown number of targets without data association using teams of mobile sensors," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 3, pp. 850–864, 2015.
- [40] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *IEEE Conference on Decision and Control*, Dec 2012, pp. 2807–2813.
- [41] G. A. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. S. Sukhatme, "Distributed data fusion for multirobot search," *Transactions on Robotics*, vol. 31, no. 1, pp. 55–66, 2015.
- [42] J. Berger and J. Happe, "Co-evolutionary search path planning under constrained information-sharing for a cooperative unmanned aerial vehicle team," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, July 2010, pp. 1–8.
- [43] M. Otte, M. Kuhlman, and D. Sofge, "Competitive target search with multi-agent teams: symmetric and asymmetric communication constraints," *Autonomous Robots*, pp. 1–24, 2017.
- [44] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 476–481 vol.1.

- [45] S. H. Arul, A. J. Sathyamoorthy, S. Patel, M. Otte, H. Xu, M. C. Lin, and D. Manocha, "Lswarm: Efficient collision avoidance for large swarms with coverage constraints in complex urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3940–3947, Oct 2019.
- [46] D. Panagou, "Motion planning and collision avoidance using navigation vector fields," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2513–2518.