Robust Environmental Mapping by Mobile Sensor Networks

Hyongju Park[1], Jinsun Liu[2], Matthew Johnson-Roberson[3] and Ram Vasudevan[1]

***Abstract*— Constructing a spatial map of environmental parameters is a crucial step to preventing hazardous chemical leakages, forest fires, or while estimating a spatially distributed physical quantities such as terrain elevation. Although prior methods can do such mapping tasks efficiently via dispatching a group of autonomous agents, they are unable to ensure satisfactory convergence to the underlying ground truth distribution when any of the agents fail. Since the types of agents utilized to perform such mapping are typically inexpensive and prone to failure, this typically results in poor overall mapping performance in real-world applications, which can in certain cases endanger human safety. To address this limitation of existing techniques, this paper presents a Bayesian approach for robust spatial mapping of environmental parameters by deploying a group of mobile robots capable of ad-hoc communication equipped with short-range sensors in the presence of hardware failures. Our approach first utilizes a variant of the Voronoi diagram to partition the region to be mapped into disjoint regions that are each associated with at least one robot. These robots are then deployed in a decentralized manner to maximize the likelihood that at least one robot detects every target in their associated region despite a non-zero probability of failure. A suite of simulation results is presented to demonstrate the effectiveness and robustness of the proposed method when compared to existing techniques.**

I. INTRODUCTION

This paper studies environmental mapping via a team of mobile robots equipped with ad-hoc communication and sensing devices which we refer to as a *Mobile Sensor Network* (MSN). In particular, this paper focuses on the challenge of trying to estimate some unknown, spatially distributed target of interest given some *a priori* measurements under the assumption that each robot in this network has limited sensing/processing capabilities. MSNs have been an especially popular tool to perform environmental mapping due to their inexpensiveness which enables large-scale deployments [1]–[7]; however, this economical price-point betrays their susceptibility to hardware failures such as erroneous sensor readings. This paper aims to develop a class of cooperative detection and deployment strategies that enable MSNs to autonomously and collectively obtain an accurate representation of an arbitrary environmental map efficiently while certifying robustness to a bounded number of sensor failures.

[1]Hyongju Park and Ram Vasudevan are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48109 USA hjcpark,ramv@umich.edu.

[2]Jinsun Liu is with the Robotics Institute, University of Michigan, Ann Arbor, MI, 48109 USA jinsunl@umich.edu.

[3]Matthew Johnson-Roberson is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, 48109 USA mattjr@umich.edu.

Few methods have been proposed to accurately perform environmental mapping using a large number of mobile robots that can guarantee robustness to hardware failures while making realistic assumptions about a MSN. For example, one of the most popular methods for addressing environmental mapping via MSNs has utilized the notion of mutual information to design controllers that follow an information gradient [2]–[6]. These approaches focus on linear dynamics and Gaussian noise models. Recently this technique was utilized to enable MSNs to estimate a map of finite events in the environments wile avoiding probabilistic failures that arose due to nearby encounters with unknown hazards [2]. The computational complexity for computing this information gradient is exponential in the number of robots, sensor measurements, and environmental discretization cells [2], [5], [6]. More problematically, the computation of the gradient requires that every robot be omniscient, i.e., have current knowledge of every other robots position and sensor measurements. For this reason, mutual information-based methods are generally restricted to small groups of robots with fully connected communication networks which has limited their potential real-world application.

To overcome this computational complexity related problem, others have focused on devising relaxed techniques to perform information gathering. For instance, some have proposed a fully decentralized strategy where the gradient of mutual information is used to drive a network of robots to perform environmental mapping [5], [6]. To improve computational efficiency they relied on a sampling technique; however this restricted their ability to perform mapping of a general complex environment instead they focus on cell environments. Others have tried to develop particle filters based techniques to enable the application of nonlinear and non-Gaussian target state and sensor models while approximating the mutual information [8]. This method is shown to localize a target efficiently. However this approach still assumes the existence of a centralized algorithm to fuse together the information from multiple sensors and to make a global decision.

Rather than rely on the information gradient, others have employed algorithm that uses information diffusion through communication network for environmental modeling [7]. By utilizing the Average Consensus filter to share information among the robots in the network, this approach is scalable to large numbers of agents, is fully decentralized, and can even work under a switching network topology as long as the network is connected; however the approach is not spatially distributed and require additional connectivity maintenance algorithm (see, e.g., [9]) to guarantee its convergence.

In this paper, we present a class of computationally efficient, scalable, decentralized deployment strategy that is robust to sensor failures. We employ classical higher order Voronoi tessellation to achieve a spatially distributed allocation of MSNs for efficient environmental mapping [10]. In particular each region from the partition is assigned to multiple robots to provide robustness to sensor failures. Although others have employed ordinary Voronoi tesselation for single robot single target assignment [3], these approaches are not guaranteed to converge to an underlying distribution in the case of even a single sensor failure [11]. To best of our knowledge, almost all studies about environmental mapping by MSNs up to date have not take into account such adversarial scenarios, nor presented performance guarantees in terms of convergence to some ground truth value, either by analysis or by numerical simulation. In addition we consider a broad class of sensor failures which are not restricted to just failures associated with proximity to a hazard.

We formulate our cost function based on a particular tessellation, and use gradient descent of the cost as our decentralized deployment strategy for the MSN. By doing so, each robot can compute gradient using merely local informations without requiring communication with a central server. In this paper, a central entity is only required to fuse and update the information gathered from MSNs, but not to generate control policies for robots as in typical mutual information gathering approaches [2], [5]. To generate an estimate for the underlying target distribution of the environment this paper employs a particle filter with low discrepancy sampling.

In addition, this paper presents a novel combined sensor model that assigns different weights to robots by taking into account the spatial relationship between robots and a target state. This detection model is based on a classical binary model that depends on the configuration of robots [12], [13]. To connect the detection model to the measurement model we rely on a nonrestrictive assumption that if a robot fails to discern one target from another; it may not provide the correct sensor reading for the target. This assumption is similar to one used in a previous approach that also built a combined sensor model that was experimentally verified with the laser range finder and a panoramic camera measurement [14]. This proposed sensor model enables one to decouple the information state from the detection task, which can make computing the gradient computationally sound with a complexity that is linear with respect to the number of sensors.

The main contributions of this paper are three-fold: First, we propose a scalable, spatially distributed, computationally efficient, decentralized controller for MSNs which can perform environmental mapping task rapidly. Second, we present a novel sensor model, to remove the computational burden of maintaining mutual information in MSNs by decoupling information gathering and detection, while ensuring satisfactory mapping performance. Finally, we adopt a higher order Voronoi tessellation for optimal robots-to-target assignment to provide robustness under a general class of sensor failures whose number is bounded.

Organization: The rest of the paper is organized as follows. Section II presents notation used in the remainder of the paper, formally defines the problem of interest. Section III presents our combined probabilistic sensor model, and the deployment strategy is formally presented in Section IV. Section V discusses an approximate belief update method via particle filters. The robustness of our deployment and effectiveness of the belief update approach is evaluated via numerical simulations in Sections VI. Finally, Section VII concludes the paper.

II. Problem description

This section presents the notation used throughout the paper, an illustrative example, and the problem of interest.

A. Notations and Our System Definition

Throughout the text, the italic bold font is used to describe random quantities, a subscript t indicates that the value is measured at time step t, and $\mathbb{Z}_{\geq 0}$ denotes nonnegative integers. Given a continuous random variable $\boldsymbol{x}$, if it is distributed according to a Probability Density Function (PDF), we denote it by $f_{\boldsymbol{x}}$. Given a discrete random variable $\boldsymbol{y}$, if it is distributed according to a Probability Mass Function (PMF), we denote it by $p_{\boldsymbol{y}}$. Consider a group of m mobile robots deployed in a workspace, i.e., ambient space, $\mathcal{Q} \subseteq \mathbb{R}^d$ where $d = 2, 3$. This paper assumes $d = 2$ though the presented framework generalizes to $d = 3$. Let $\mathbb{S}^{d-1} = \{s \in \mathbb{R}^d \mid \|s\| = 1\}$ be a unit circle/sphere, then the state of m robots is the set of locations and orientations at time t, and it is represented as an m-tuple $x_t = (x_t^1, \ldots, x_t^m)$, where $x_t^i \in \mathcal{Q} \times \mathbb{S}^{d-1}$. The state of robots are assumed completely *known*. We denote by the set $x_{0:t} \coloneqq \{x_0, \ldots, x_t\}$ the robot states up to time t.

The group of robots follow a way-point-based, discretized, deterministic kinematic motion model:

$$x_{t+1} = x_t + u_t, \quad t \in \mathbb{Z}_{\geq 0} \tag{1}$$

where $u_t \in \mathcal{U} \subseteq \mathbb{R}^d$ is the control. Similarly, let $u_{0:t}$ be the sequence of control policies up to time t. We define a *target* to be a physical object or some measurable quantity spatially distributed over a bounded domain. Let $\boldsymbol{z}$ be the *target state* which is a random vector. The target state consists of location, $\boldsymbol{q} \in \mathcal{Q}$, and information state (i.e., quantitative information about the target), $\boldsymbol{I} \in \mathcal{I} \subseteq \mathbb{R}$ where we let $\mathcal{I} = [I_{\min}\, I_{\max}]$. The Cartesian product $\mathcal{Z} = \mathcal{Q} \times \mathcal{I}$ is the *target state space*. Let $\boldsymbol{y}_t = (\boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^m)$ be binary random m-tuple which indicates the observation made by m robots at time step t where $\boldsymbol{y}_t^i \in \{0, 1\}$ for each i. Let the set $\boldsymbol{y}_{1:t} \coloneqq \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ denote observations made by robots up to time t. Let $\boldsymbol{y}_D$ a binary random tuple of size m where each $\boldsymbol{y}_D^i \in \{0, 1\}$ which denotes the detection by ith robot. Finally, we let $\mathcal{F} \subsetneq \{1, \ldots, m\}$ be the index set of robots whose sensors have failed.

Example 2.1 (Airborne LiDARs for DEM generation): Consider a group of autonomous aerial vehicles trying to acquire an accurate Digital Elevation Model (DEM)[1] of some bounded region using airborne LIDAR measurements. The first problem is: given a finite number of vehicles, how to deploy those vehicles such that the probability that robots will fail to targets dispersed over the region of interest stays minimum. Also, it may be possible that LIDAR measurements from a part of the fleet are corrupt or unreliable which can potentially degrade the quality of DEM. Taking such adversarial scenarios into account, it is crucial to design a deployment strategy along with an underlying sensor model, which can guarantee worst-case optimal target detection performance. Finally, it is also important to process the measurement to update the belief on target state over time in efficient manner such that, within a reasonable time, the autonomous fleet can obtain an accurate DEM model.

B. Deployment strategy for the worst-case optimal target detection under sensor failures

For a given target located at $q \in \mathcal{Q}$, let $p(\boldsymbol{y}_D^i = 0 \mid x^i, \boldsymbol{q} = q)$ be the probability that ith robot fails to detect target located at q, and $\boldsymbol{y}_D^i = 1$ otherwise. In a similar manner, $p(\boldsymbol{y}_D = \boldsymbol{0} \mid x, \boldsymbol{q} = q)$ is the probability of a joint event that a group of m robots fail to detect the target at q. For the case when sensors with identifies $\mathcal{F}$ have failed, our aim is to find an optimal configuration which solves:

$$x^\star = \arg\min_{x \in \mathcal{Q}^m} p(\boldsymbol{y}_D = \boldsymbol{0} \mid x) \tag{2}$$

subject to

$$p(y_D = \boldsymbol{0} \mid x, \boldsymbol{q} = q, \mathcal{F}) < 1, \quad q \in \mathcal{Q}. \tag{3}$$

In other words given the sensor failure $\mathcal{F}$, for each target, there must be at least one robot that can detect it, which is an essential constraint for the environmental mapping task[2]. The target distribution of $\boldsymbol{q}$ has been marginalized out. Unfortunately, obtaining the global solution of even the outer minimization problem can be proven to be NP-Hard by reduction from a simpler static locational optimization problem, namely, *m-median problem*[3]. To overcome the computational complexity, we will consider a gradient descent (greedy) approach where control policy at each time step minimizes the missed-detection probability of targets by robots at their future locations (one-step lookahead). We will utilize the higher order Voronoi tessellation [10] for robot target assignment, and will show that the solution under such assignment solves (2).

[1] A digital elevation model (DEM) is a digital 3D model of a terrain's surface created from terrain elevation data.

[2] This paper considers an average-case optimal solution with respect to sensor failures. For our purpose, the worst-case optimal (minimax) solution can, in fact, be too conservative.

[3] m-median problem is one of the popular locational optimization problem where the objective is to locate m facilities to minimize the distance between demands and the facilities given uniform prior. The problem is NP-Hard in general graph (not necessarily a tree).

C. Combined sensor model

Here, we make a simple assumption that given a target, a sensor can correctly measure the target only if the sensor can detect target (i.e., the sensor can discern the target from others.) *a priori*. We further assume that, if a sensor can detect a target, a measurement of the target by the sensor is simply corrupted by noise (e.g., Gaussian). On the other hand, of a robot fail to detect a target, the measurement given the target may be unreliable and should not be considered equally as a correct measurement. These assumptions have turned out to be non-restrictive according to the work of [14] in which authors has experimentally verified a similar combined sensor model on a mobile robot using laser range finder and a panoramic camera measurement.

D. Evaluation of mapping performance

We derive particle filter to recursively update approximate beliefs on a particular unknown environment. Let $\hat{b}_t$ represent a the approximate posterior probability distribution of the target state at time $t \in \mathbb{Z}_{\geq 0}$, the initial belief $\hat{b}_0$ is assumed to be uniform density if no prior information on the target is available, and let $\hat{b}^\star$ be the PMF estimate of the *true posterior belief*[4] To this end, we quantify the difference between the true posterior belief, $b^\star$ and our method via the Kullback-Leibler (K-L) divergence. We demonstrate via a suite of numerical simulations in Section VI that for a given $\epsilon > 0$ and $\mathcal{F} \neq \emptyset$, there is reasonably small $T > 0$ such that if robots use the proposed deployment strategy, $t > T$ implies $D_{\mathrm{KL}}(\hat{b}_t || \hat{b}^\star) < \epsilon$.

III. Probabilistic Range-limited Sensor Model

In this section, we present our combined sensor model. Each mobile robot is equipped with a *range-limited sensor* that can measure quantitative information from afar and a *radio* to communicate with other nodes to share its belief. Each range sensor measurement of a target is corrupted by noise, and the measurement is valid only if the target has been detected. This combined sensor model joins the generic noisy sensor model with the binary detection model [12], [13]. In fact, this combined sensor model has been experimentally validated during an object mapping and detection task using a laser scanner [14]. We postulate that this model is general enough to model other range-limited sensors as well; as long as the sensor is capable of distinguishing the target from the environment, and has uniform sensing range. A few example sensors satisfying these characteristics are 360-degree camera, wireless antenna, Gaussmeter, heat sensor, olfactory receptor, etc. While performing the detection task, we assume each sensor returns a 1 if a target is detected or 0 otherwise. The ability to detect a target for each i^{th} robot is a binary random variable $\boldsymbol{y}_D^i$ with a distribution that depends on the relative distance between the target and robot. This binary detection model, however, does not account for false positive or negatives. For example, the probability of the

[4] We assume for now that the true posterior target distribution can be obtained, e.g., via exhaustive search and measurements made by a MSN on the target state.

event that all m sensors with configuration x_t fail to detect the target located at $q \in \mathcal{Q}$ is:

$$\begin{aligned} & p_{\boldsymbol{y}_{D,t}|x_t,\boldsymbol{z}}\left(\boldsymbol{y}_{D,t} = \mathbf{0} \mid x_t, \boldsymbol{z} = (q, I)\right) \\ &= \prod_{i=1}^{m} p_{\boldsymbol{y}^i_{D,t}|x_t,\boldsymbol{q}}\left(\boldsymbol{y}^i_{D,t} = 0 \mid x_t, \boldsymbol{q} = q\right), \end{aligned}$$

where $\mathbf{0}$ is a m-tuple of zeros. For measuring a quantity of interest from a given environment, we consider a generic, noisy sensor model, where each sensor reports binary output given a target state consists of information and location. The likelihood function at time t is:

$$p(\boldsymbol{y}_t = \mathbf{1} \mid x_t, \boldsymbol{z} = (q, I)), \tag{4}$$

which is the probability that ith robot measured the target with intensity value of I at location q, i.e., positive measurement. A general example of the likelihood function is a Gaussian, $\omega\mathcal{N}(I, I^\star, \sigma_I^2)$ where $I^\star$ the ground truth intensity value at q, σ_I^2 is the variance of the intensity at the target located at q, and ω is a normalization constant. Note that since the observations made by m robots are independent,

$$p(\boldsymbol{y}_t = \mathbf{1} \mid x_t, \boldsymbol{z} = (q, I)) = \prod_{i=1}^{m} p(\boldsymbol{y}^i_t = 1 \mid x_t, \boldsymbol{z} = (q, I)),$$

or (4) can be obtained via other distributed sensor fusion techniques (see e.g., [15]). In our sensor model, we assume that the random vector $\boldsymbol{y}$ depends on $\boldsymbol{y}_D$ which is a random m-tuple corresponds to detection by m robots such that $\boldsymbol{y}_D = (\boldsymbol{y}^1_D, \ldots, \boldsymbol{y}^m_D)$ where $\boldsymbol{y}^i_D \in \{0, 1\}$ for all i when conditioned on $x_t, \boldsymbol{z}$, so that the conditional PDF can be computed as:

$$\begin{aligned} & f_{\boldsymbol{y}_t|\boldsymbol{z},x_t}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t) \\ &= f_{\boldsymbol{y}_t|\boldsymbol{z},x_t,\boldsymbol{y}_{D,t}}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t, \boldsymbol{y}_{D,t} \neq \mathbf{0}) \\ &\quad \times p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t}(\boldsymbol{y}_{D,t} \neq \mathbf{0} \mid z, x_t) \\ &+ f_{\boldsymbol{y}_t|\boldsymbol{z},x_t,\boldsymbol{y}_{D,t}}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t, \boldsymbol{y}_{D,t} = \mathbf{0}) \\ &\quad \times p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid z, x_t), \end{aligned}$$

where $\boldsymbol{y}_D \neq \mathbf{0}$ means there is $j \in \{1, \ldots, m\}$ such that $\boldsymbol{y}^j_D = 1$ and $\boldsymbol{y}_D = \mathbf{0}$ means $\boldsymbol{y}^j_D = 0$ for all $j \in \{1, \ldots, m\}$. If target is missed-detected, i.e., $\boldsymbol{y}_{D,t} = \mathbf{0}$, the measurement is taken as random and the likelihood function is modeled by uniform distribution, i.e.,

$$\begin{aligned} & f_{\boldsymbol{y}_t|\boldsymbol{z},x_t,\boldsymbol{y}_{D,t}}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t, \boldsymbol{y}_{D,t} = \mathbf{0}) \\ &= f_{\boldsymbol{y}_t|\boldsymbol{z},x_t,\boldsymbol{y}_{D,t}}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t, \boldsymbol{y}_{D,t} = \mathbf{0}) = I_{\text{range}}^{-1} \end{aligned}$$

where $I_{\text{range}} := I_{\max} - I_{\min}$. By the law of total probability,

$$\begin{aligned} & f_{\boldsymbol{y}_t|\boldsymbol{z},x_t}(\boldsymbol{y_t} = \mathbf{1} \mid z, x_t) \\ &= (1 - \underbrace{p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid z, x_t))}_{(\star)\text{ the probability of missded detection}} \\ &\quad \underbrace{\times f_{\boldsymbol{y_t}|\boldsymbol{z},x_t\boldsymbol{y}_{D,t}}(y_t \mid z, x_t, \boldsymbol{y}_{D,t} \neq \mathbf{0})}_{\text{the likelihood of reliable measurements}}, \\ &+ I_{\text{range}}^{-1} p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid z, x_t). \end{aligned}$$

For the given target $z \in \mathcal{Z}$, by minimizing $(\star)$, one can ensure that the reliable measurements on the target state has been given more weight than the unreliable ones.

IV. Deployment strategy

This section presents a class of deployment strategies for target detection capable of providing relative robustness. At each time, m robots move to new locations so as to minimize the missed-detection probability to promote the next observations. Since the set of robots with faulty sensors is unknown, we chose not to pose our problem to deal with the worst-case sensor failure scenarios which could be too conservative (e.g., the probability that multiple sensors fail at the same time is low). Instead, we adopt a provably optimal robot–target assignment method which can ensure that every target will be detected by at least one robot. This so called partitioned-based deployment is common to multi-robot coverage problems [11], [16]–[18]. The most popular one is based on the Voronoi tessellations (see e.g., [16], which we call a *non-robust deployment*). There are, in fact more general methods, which partition the workspace into l regions and assign $k \in \{1, \ldots, m\}$ robots each region (note that if $k = m$, the method becomes *centralized*) [11]. By doing so, one can ensure that each target has a chance to be detected by at least one of the k sensors. This approach, which we call the *robust deployment*, can provide relative robustness by varying the value of k from 1 to m.

A. The higher-order Voronoi Partition for Robust deployment

We will utilize the higher order Voronoi Tessellation to achieve robust robot–target assignment. Given the maximum possible number of sensor failures, let's say $f = |\mathcal{F}|$, as noted before, we want to ensure that at least one robot can detect a target. One way to do so is using k-coverage method [19], where each target is being covered by at least k sensors. Also, we may use the higher order Voronoi tessellation, which, for a given number of sensors (generators), assigned exactly k number of sensors for every region from a partition. For both approaches if $k \geq f + 1$, the constraint is always satisfied. Since we have bounded sensors available, the second approach is more desirable, and we will present the method in this study. Consider m sensors and a workspace partition of $\mathcal{Q}$ into l disjoint regions, W such that $W = (W^1, \ldots, W^l)$, where $\cup_i W^i = \mathcal{Q}$, and $W^i \cap W^j = \emptyset$ for all i, j pairs with $i \neq j$. Suppose the target location is a random variable $\boldsymbol{z}$ with PDF $\phi : \mathcal{Q} \to \mathbb{R}_{\geq 0}$. For a given target $q \in \mathcal{Q}$, we define the probability that a sensor located at x can detect target, by using a real-valued function $h(\|q - x^i\|)$ as a probability measure[5], which is assumed to decrease monotonically as a function of the distance between the target and the i^{th} sensor. Consider a bijection ${}^kG$ that maps a region to a set of k-points where the pre-superscript k explicitly states that the region is mapped to exactly k points. Additionally we make the following definitions:

[5]For the numerical simulations purpose, we further assume that $h(\cdot)$ is continuously differentiable function non-increasing on its domain, and the image of h must be in $[0, 1]$ for it to be a probability measure.

Definition 4.1 (An Order−k Voronoi Partition [10]): Let x be a set of m distinct points in $\mathcal{Q} \subseteq \mathbb{R}^d$. The *order-k Voronoi partition of $\mathcal{Q}$ based on x*, namely ${}^kV$, is the collection of regions that partitions $\mathcal{Q}$ where each region is associated with the k nearest points in x.
We also define another bijection ${}^kG^\star$ that maps a region to a set of k *nearest* points (out of x) to the region. The total probability that all m sensors fail to detect a target drawn by a distribution ϕ from $\mathcal{Q}$ is:

$$\int_{\mathcal{Q}} p_{\boldsymbol{y}_D|x,\boldsymbol{q}}\left(\boldsymbol{y}_D = \mathbf{0} \mid x, \boldsymbol{q} = q\right) \phi(q)\, dq. \tag{5}$$

By substituting $\mathcal{Q}$ with the workspace partition W, and $p_{\boldsymbol{y}_D|x,\boldsymbol{q}}\left(\boldsymbol{y}_D = \mathbf{0} \mid x, \boldsymbol{q} = q\right)$ with h, we have

$$\begin{aligned} &H(x, W, {}^kG) \\ &= \sum_{j=1}^{l} \int_{W^j} \left(\prod_{x^i \in {}^kG(W^j)} \left(1 - h\left(\left\|q - x^i\right\|\right)\right) \right) \phi(q)\, dq \end{aligned} \tag{6}$$

where we note again that the joint missed-detection events are conditionally independent, if conditioned on x. In fact, the order-k Voronoi tessellation is the optimal workspace partition which minimizes H for each choice of x and k:

Theorem 4.1 ([18]): For a given x and k, $H(x, {}^kV, {}^kG^\star) \leq H(x, W, {}^kG)$ for all W, ${}^kG$.
Note that the order-k Voronoi partition V_k, along with the map $G_k^\star$ are uniquely determined given x, ϕ, and $\mathcal{Q}$.

Our proposed sensor model utilizes the deterministic workspace partitioning method based on the order-k Voronoi partition toward the probabilistic sensor model presented in Section III. In addition, we introduce additional hard constraint for the model, namely the *effective sensing range*, $r_{\text{eff}} > 0$, to take into account the fact that each sensor has its own maximum sensing range. For a given k, and the target $z = (q, I)$, our range-limited binary detection model is:

$$\begin{aligned} &p_{\boldsymbol{y}^i_{D,t}|\boldsymbol{q},x_t}\left(\boldsymbol{y}^i_{D,t} = 1 \mid, x_t, \boldsymbol{q} = q\right) \\ &= \begin{cases} h\left(\left\|q - x^i_t\right\|\right) & \text{if } q \in {}^kG^\star_t(x^i_t) \cap \mathcal{B}(x^i_t, r_{\text{eff}}), \\ 0, & \text{otherwise}, \end{cases} \end{aligned}$$

where $\mathcal{B}(x, r)$ is an open ball with radius r centered at x.

B. Gradient algorithm

This section will present gradient-decent-based deployment strategy. Given the current configurations, robots compute and move their next way-points, and the posterior belief is updated at the new locations given the information collected from their sensors. By using $f_{\boldsymbol{q}}$, and (6), for a given x_{t-1} at time $t-1$, we want to obtain the next way-point $x_t^\star$ which solves

$$\begin{aligned} x_t^\star \leftarrow \arg\min_{x_t} \Bigg\{ &\mathcal{L}(x_t) := \\ &\sum_{j=1}^{l} \int_{W_t^j} \prod_{x_t^i \in {}^kG^\star(W_t^j)} \left(1 - h\left(\left\|q - x^i_t\right\|\right)\right) f_{\boldsymbol{q}}(q)\, dq \Bigg\}. \end{aligned} \tag{7}$$

Note that for a given k, by substituting $f_{\boldsymbol{q}}$ with ϕ and x with x_t, (7) takes the identical form as $H(x, W, G^k)$ which was previously defined in (6). If h is differentiable, our deployment strategy can use the gradient $\nabla\mathcal{L}(x_t) = \left[\frac{\partial\mathcal{L}(x_t)}{\partial x_1^t}, \ldots, \frac{\partial\mathcal{L}(x_t)}{\partial x_m^t}\right]$ where for each i,

$$\begin{aligned} \frac{\partial\mathcal{L}(x_t)}{\partial x^i_t} = &- \sum_{\substack{j \in \{1,\ldots,l\}: \\ W_t^j \in {}^kG^{\star-1}(x_t^i)}} \int_{W_t^j} \frac{\partial h(\left\|q - x^i_t\right\|)}{\partial x^i_t} \\ &\times \prod_{\substack{l \in \{1,\ldots,m\}: \\ x_t^l \in {}^kG^\star(W_t^j), l \neq i}} \left(1 - h\left(\left\|q - x^i_t\right\|\right)\right) f_{\boldsymbol{q}}(q)\, dq, \end{aligned}$$

to find the desirable control policy of the robots as described in Algorithm 1. Algorithm 1 uses coordinate gradient descent in cyclic fashion to obtain a sub-optimal solution, namely, $\hat{u}_t$ for each time t. Algorithm 1 can be shown to be convergent

Algorithm 1: Gradient Algorithm

```
Input: L_k, x̂_t, ε > 0
Output: û_t
k ← 0, Δ ← ε
while Δ > ε do
    foreach i ∈ {1, ..., m} do
        x^i_{t,k+1} ← x^i_{t,k} − α^i_{t,k} ∇_i L_k(x_{t,k})
        // α^i_{t,k} is a step-size obtained using a line search method
    Δ ← L_k(x_{t,k}) − L_k(x_{t,k+1})
    k ← k + 1
x̂*_t ← x_{t,k}, û_t ← x̂*_t − x̂_t
return û_t
```

via Invariance Principle. The theorem along with the formal proof is contained in our previous paper [18].

V. Implementation: Environmental Mapping

In this section, we first introduce Bayesian filtering equations for our particular target distribution, and then present a particle filter to reduce the complexity of the map construction process.

A. Recursive Bayesian Filter

We present a brief overview of the Bayesian filter, and the derivation of the filtering equations for our primary goal: environmental mapping by m robots. Recall that $b_t(z)$ represent a *belief* on target state—the posterior probability distribution of the target state described by a random vector $\boldsymbol{z} \in \mathcal{Z}$—at time $t \in \mathbb{Z}_{\geq 0}$. In a similar manner, the belief of target information state I given the target located at q is given by:

$$b_t(I \mid \boldsymbol{q} = q) = f_{\boldsymbol{I}|b_0,x_{0:t}\boldsymbol{y}_{1:t},\boldsymbol{q}}\left(I \mid b_0, x_{0:t}, y_{1:t}, \boldsymbol{q} = q\right) \tag{8}$$

where we denote the initial belief on target state by b_0. The belief on the complete target state $\boldsymbol{z}$ is:

$$b_t(z) = f_{\boldsymbol{z}|b_0,x_{0:t},\boldsymbol{y}_{1:t}}\left(z \mid b_0, x_{0:t}, y_{1:t}\right) = b_t(I \mid \boldsymbol{q} = q) f_{\boldsymbol{q}}(q). \tag{9}$$

In our problem, the observation $\boldsymbol{y}_t$ is conditionally independent of b_0, $\boldsymbol{y}_{1:t-1}$, and $x_{0:t-2}$ when it is conditioned on $\boldsymbol{z}$

and x_t. Applying *Bayes' Theorem*, (8) becomes:

$$b_t(I \mid \boldsymbol{q}=q) = \eta_t\, f_{\boldsymbol{y}_t|\boldsymbol{z},x_t}\left(\boldsymbol{y}_t=\mathbf{1} \mid \boldsymbol{z}=(I,q),x_t\right) b_{t-1}(I \mid \boldsymbol{q}=q) \quad (10)$$

where $\eta_t := \left(f_{\boldsymbol{y}_t|\boldsymbol{q},b_0,x_t}\left(\boldsymbol{y}_t=\mathbf{1} \mid \boldsymbol{q}=q,b_0,x_t\right)\right)^{-1}$ is a *normalization constant*. By joining the (9) and (10), one can obtain a simplified form of the filtering equation:

$$\begin{aligned} b_t(z) &= \eta_t\, f_{\boldsymbol{y}_t|\boldsymbol{z},x_t}\left(\boldsymbol{y}_t=\mathbf{1} \mid z,x_t\right) b_{t-1}(z) \\ &= \left(\prod_{i=1}^{t} \eta_i f_{\boldsymbol{y}_i|\boldsymbol{z},x_i}\left(\boldsymbol{y_i}=\mathbf{1} \mid z,x_i\right)\right) b_0(z). \end{aligned}$$

B. Belief Approximation via SIR Particle Filter

For our numerical simulations, we consider a low discrepancy sampling method, namely, Halton-Hammersley sequence, to sample continuously distributed targets in $z \in \mathcal{Z}$. This approach has been used for sampling-based algorithms for robot motion planning [20]. We consider Sequential Importance Resampling (SIR) [21] for the particle filtering process. For a given distribution on target locations, $f_{\boldsymbol{q}}(q)$, at each time t, based on the observations, the locations belief hypothesis is populated for N_1 samples initially generated with Halton-Hammersley sequence, $q^1,\dots,q^{N_1}$ where for each $i \in \{1,\dots,N_1\}$, $j \in \{1,\dots,N_2\}$,

$$\widetilde{w}_t^{ij} \propto f_{\boldsymbol{y}_t|\boldsymbol{z}_t,x_t}(\boldsymbol{y}_t=\mathbf{1} \mid \hat{x}_t,\, \boldsymbol{z}=(q^i,I^{ij})).$$

In a similar manner, for each sample q^i the information belief hypothesis is populated for N_2 samples from $\mathcal{I}$ initially generated by the Halton-Hammersley sequence. If we let $z_t^{ij} := (q^i, I_t^{ij})$, then the collection of $N := N_1 \times N_2$ tuples—where each tuple is a particle-weight pair—is

$$\left\{\left\{\left(z^{i1},\widetilde{w}_t^{i1}\right),\dots,\left(z^{iN_2},\widetilde{w}_t^{iN_2}\right)\right\}_{i\in\{1,\dots,N_1\}}\right\}$$

where for each t and $i=1,\dots,N_1$, $\sum_{j=1}^{N_2}\widetilde{w}_t^{ij}=1$. After resampling and normalizing, the approximate posterior belief becomes

$$\hat{b}_t(z) = \sum_{k=1}^{N} w_t^k \delta(z-z^k)$$

which is a form of discrete random measure where the $w_t^1,\dots,w_t^N$ are resampled, normalized weight such that $\sum_{k=1}^{N} w_t^k = 1$, and $\delta(z-z^k)$ is Dirac-delta function evaluate at z^k. The whole filtering process is depicted in Algorithm 2. Note that as discussed in previous studies [22], our particle filter uses standard re-sampling scheme to ensure the convergence of the mean square error toward zero with a convergence rate of $1/N_2$ for all $q \in \mathcal{Q}$.

VI. Numerical Simulations

This section presents a suite of numerical simulations to validate both our sensor model and deployment strategy. In particular, we will focus on comparing our approach to popular state-of-the-art approaches under various sensor failure scenarios.

Algorithm 2: Filtering Algorithm

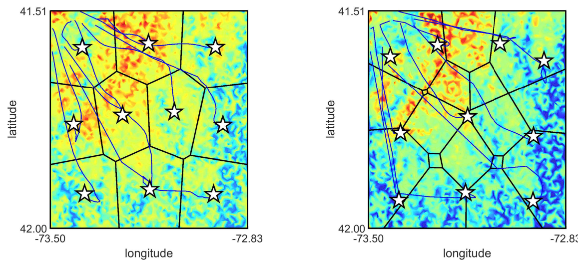
```
Input: b̂_{t-1} = {z^l, w^l_{t-1}}^N_{l=1}, y_t, x̂_t, I_range
Output: b̂_t
// SIR Particle Filter
// 1) Update using the observation model
foreach i ∈ {1,...,N_1} do
    foreach j ∈ {1,...,N_2} do
        w̃^{ij}_t ← p_{y_{D,t}|q,x̂_t}(y_{D,t} = 0 | q = q^i, x̂_t)(I^{-1}_range −
          w^{ij}_{t-1} f_{y_t|z,x̂_t,y_{D,t}}(y_t = 1 | z = z^{ij}_t, x̂_t, y_{D,t} ≠
          0)) + w^{ij}_{t-1} f_{y_t|z,x̂_t,y_{D,t}}(y_t = 1 | z, x̂_t, y_{D,t} ≠ 0)
// 2) Resample and Normalize
{w^l_t}^N_{l=1} ← Resample({w̃^l_t}^N_{l=1}, {w^l_{t-1}}^N_{l=1})
return b̂_t ← {z^l, w^l_t}^N_{l=1}
// Low Variance Resampling [23]
function Resample({w̃^l_t}^N_{l=1}, {w^l_{t-1}}^N_{l=1})
    forall i ∈ {1,...,N} do
        w̄^i_t ← (w̃^i_t · w^i_{t-1}) / (Σ^N_{i=1} w̃^i_t · w^i_{t-1})
    foreach i ∈ {1,...,N_1} do
        δ ← rand((0; N_2^{-1}))
        cdf ← 0, k ← 0, c_j ← [] for all j
        for j = 0, j < N_2 do
            u ← δ + j · N_2^{-1}
            while u > cdf do
                k ← k + 1
                cdf ← cdf + w̄^{ik}_t
            c_{j+1} ← k
        for j = 1; j ≤ N_2 do
            w^{ij}_t ← c_j / N_2
    return b̂_t = {z^l, w^l_t}^N_{l=1}
```

Simulation settings: We consider $\mathcal{Q}$ be a rectangular space $[42.00, 41.51] \times [-73.49, -72.83]$ in $\mathbb{R}^2$, some mountain area in Connecticut, U.S.A, where each coordinate corresponds to latitude and longitude, respectively. We let $\mathcal{I} = [-1000, 4000]$ range of elevation in feet, and $r_{\text{eff}} = 2.89$ miles. Targets are uniformly distributed over $\mathcal{Q}$, and the initial expected value for initial target information over $\mathcal{Q}$ is depicted in Fig. 4(right). The information vector ranges from -1000 to 4000, i.e., $\mathcal{I} = [-1000, 4000]$. It is assume that the robots have no prior knowledge of the target information. A number of particles used for the SIR filter is $N = N_1 \times N_2 = 5000 \times 100$. We consider Gaussian kernels for the probability distributions of both the perception model, and the detection model. The value of the equipped noisy sensor's covariance matrix is $\Sigma_I = 0.5\mathbf{I}$, and the binary sensor's covariance is $\Sigma_B = 0.04\mathbf{I}$ where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is an identity matrix. In our simulation, we will compare the three methods summarized in Table I. Note that the three methods presented here are not exactly same as those found from the reference therein, nevertheless, we postulate that the results will be comparable due to the similarities of the ideas.

Convergence of our deployment algorithm: First, the behavior of the deployment strategy is discussed. Given the initial uniform prior belief and an initial configuration at $t = 0$ (upper-left corner), three algorithms, summarized in

TABLE I: Summary of deployment methods considered in current section:

algorithm type:	gradient computation	related studies
non-robust	fully decentralized	[3], [16]
robust ($k = 2$)	decentralized	current paper
max. information gain	centralized	[1]–[3], [5], [7]



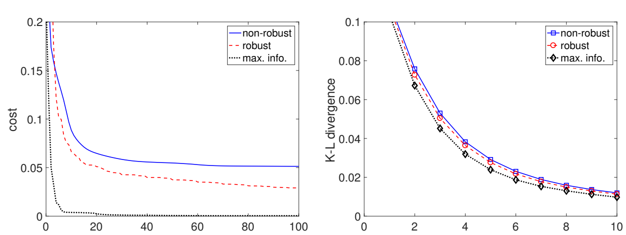
(a) (b)

Fig. 1: One-time deployment with non-robust (left) and robust ($k = 2$) (right) overlaid with map at $t = 1$ (lines: gradient descent flow, stars: positions of robots, polygons: partition).

Table I, were tested. Fig. 1 shows the configurations after $t = 1$ with (a) non-robust method and (b) robust method ($k = 2$). Fig. 2(a) compares the convergence speeds and the cost changes between three methods.

Environmental mapping/filtering performance without sensor failure: Next, we present the evolution of the object map given the uniform, initial map with successive positive observations, each followed by filtering process and the gradient descent strategy. Fig. 2(b) compares the K-L divergence values between different strategies during the filtering process. While the maximum information gain approach shows the best result, other methods also shows competitive mapping performance relative to the ground truth.

Robustness to sensor failure: We will present a number of examples with sensor failures where robust method become more appealing than non-robust method. The number of sensors that have failed was varied by $\mathcal{F} = \{1\}, \{1, 2\}, \{1, 2, 3\}$. Results for robots configuration and target distributions after 10^{th} step with non-robust and robust methods in the case when $\mathcal{F} = \{1\}$ are shown in Fig. 3. Fig. 4 shows the time evolutions of root-mean-square error (RMSE) between constructed map and the ground truth map for both non-robust method (top) and robust method (bottom) when $\mathcal{F} = \{1, 2, 3\}$. Fig. 5 compares the K-L divergence between difference methods when $|\mathcal{F}|$ was varied between 1



(a) (b)

Fig. 2: Convergence test for one-time deployment with different methods.

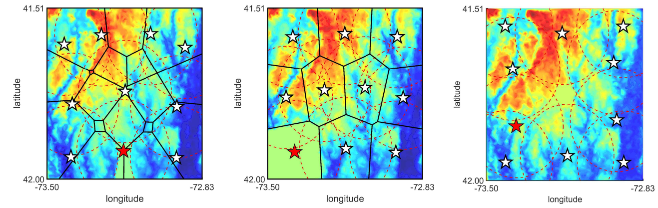


Fig. 3: The belief at $t = 10$ with non-robust method (left) and robust method (middle), and max. information method (right) Also showing robot configurations (stars, where red stars are robots whose sensors have failed), effective sensing range (dashed line), and partition (solid lines).

and 3. As can be seen from Fig 3–5, the map retrieved by the proposed method $k = 2$ is more robust to the sensor failure compared to that obtained with the non-robust method. In Fig 3, both middle and right, the unmapped area is owe to the limited sensing range.

Statistical results with different initial conditions/faults compositions: Statistical results shows that our method can be used to estimate arbitrary target distribution given randomly chosen initial configuration, with different fault compositions reasonably well. Fig. 6 shows a distribution of K-L divergence values at the $t = 10$ for a given 100 random initial configurations with randomly sampled faults whose numbers are between 1 and 5.

Scalability of our method: We also conducted a series of simulations and have validated that the previous results with 10 robots can be generalize to larger number of robots Fig. 2 shows an example with 100 robots (with 10 sensors failures) where the robust method outperforms the non-robust method.

Remark: It is not surprising to see from the simulation results that the maximum information gain approach has better robustness performance against sensor failures than other methods; however this is due to the presence of central information fusion server, and relatively large communication load. Thus, it is unfair to compare robustness between centralized (max. information gain) and decentralized (non-robust, robust) methods.

VII. Conclusions and Future Work

This paper presents a general deployment strategy for autonomous fleet to maximize the recovery of environmental map over a bounded space, robots to sensor failures. It is expected that our method will fail if there is not enough number of mobile agents having sufficiently long effective sensing ranges compared to the workspace size. One of our future works is, therefore, to employ multi-agent patrolling [24] or sweep coverage [25] algorithms or to resolve such problems where there may not be enough sensors to cover the whole target space. Also, as reported in the literature [14], our combined sensor model has been adopted to emulate the real-world laser scanner's behavior, nevertheless, it is one of our future works to conduct extensive real world multi-robot experiments for further validation of our range sensor model. Lastly, we assumed in this study that the beliefs are shared between robots such that both tasks of information

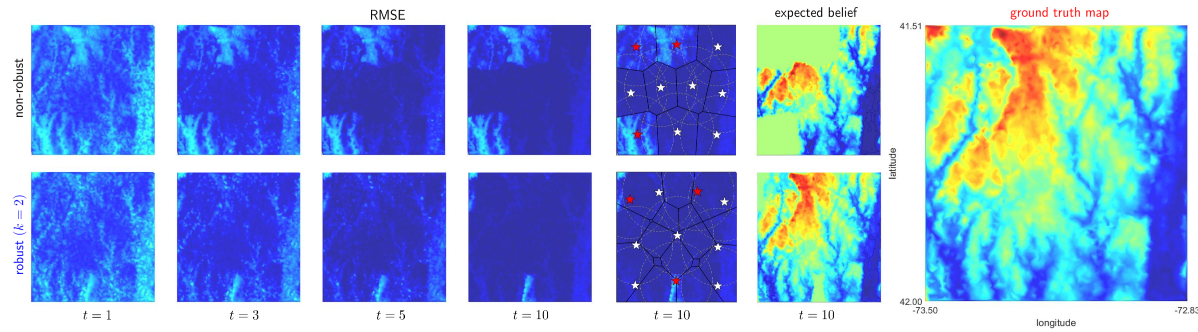


Fig. 4: Time evolutions of root-mean-square error (RMSE) (in ft) between constructed map and the ground truth map for non-robust method (top) and robust method (bottom) when $\mathcal{F} = \{1, 2, 3\}$.

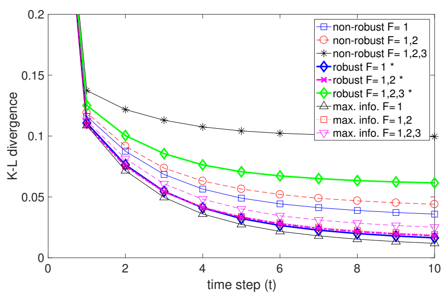


Fig. 5: Comparison of K-L divergence from the actual distribution between different methods during belief propagation when part of the sensor fail.

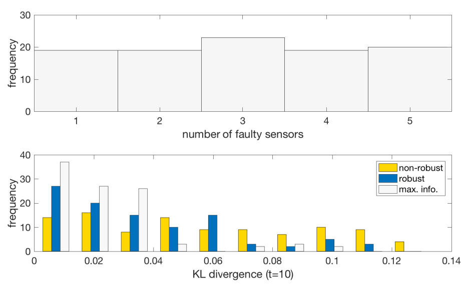
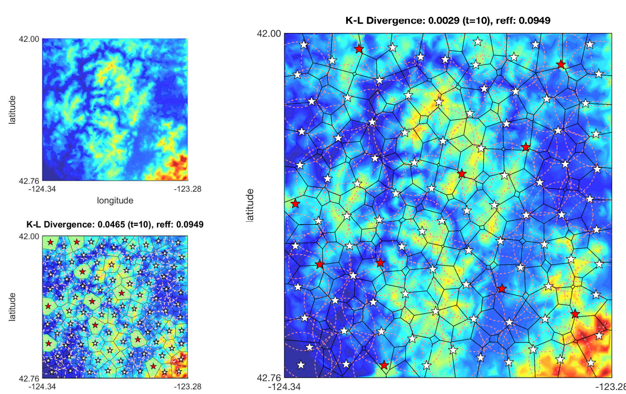


Fig. 6: Robustness test with 100 test dataset, distribution of $|\mathcal{F}|$ (top), K-L divergence at $t = 10$ (bottom).



(a) (b)

Fig. 7: Simulations with 100 robots estimating an elevation map of a mountain area (Oregon USA) where 10 sensors have failed (a)-top: ground truth, configuration and partition overlaid over the expected belief after $t = 10$ [(a)-bottom: non-robust method, (b) robust method].

gathering, propagating and approximating belief require a central entity. In the future, we will explore how to devise distributed communication protocol to enable distributed belief estimation.

REFERENCES

[1] D. Connor, P. Martin, and T. Scott, "Airborne radiation mapping: overview and application of current and future aerial systems," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5953–5987, 2016.

[2] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *Robotics Research*. Springer, 2017, pp. 455–472.

[3] R. A. Cortez, H. G. Tanner, R. Lumia, and C. T. Abdallah, "Information surfing for radiation map building," *International Journal of Robotics and Automation*, vol. 26, no. 1, p. 4, 2011.

[4] C. D. Pahlajani, I. Poulakakis, and H. G. Tanner, "Networked decision making for poisson processes with applications to nuclear detection," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 193–198, 2014.

[5] 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.

[6] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.

[7] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[8] J. A. Hoffman, J. R. Cunningham, A. J. Suleh, A. Sundsmo, D. Dekker, F. Vago, K. Munly, E. K. Igonya, and J. Hunt-Glassman, "Mobile direct observation treatment for tuberculosis patients: a technical feasibility pilot using mobile phones in nairobi, kenya," *American journal of preventive medicine*, vol. 39, no. 1, pp. 78–80, 2010.

[9] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[10] M. I. Shamos and D. Hoey, "Closest-point problems," in *Foundations of Computer Science, 1975., 16th Annual Symposium on*. IEEE, 1975, pp. 151–162.

[11] S. Hutchinson and T. Bretl, "Robust optimal deployment of mobile sensor networks," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, p. 671676.

[12] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors part i. fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, 1997.

[13] P. M. Djuric, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2229–2238, 2008.

[14] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3777–3784.

[15] A. W. Stroupe, M. C. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 1092–1098.

[16] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, p. 243255, 2004.

[17] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[18] H. Park and S. Hutchinson, "Robust optimal deployment in mobile sensor networks with peer-to-peer communication," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2144–2149.

[19] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 144–158.

[20] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[21] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[22] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on signal processing*, vol. 50, no. 3, pp. 736–746, 2002.

[23] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[24] D. Portugal and R. Rocha, "A survey on multi-robot patrolling algorithms," in *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2011, pp. 139–146.

[25] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3462–3468.