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]–[5]; 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 [3], [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], [4]. 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 [4] 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 [7]. 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 algorithms that decrease the variance of the state estimate of the environment model to perform tasks such as information diffusion (**Ram: are they performing environmental mapping?**) [5]. 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, since the focus is on information diffusion rather than building a

spatially distributed map, this approach may not be ideal for environmental mapping (**Ram: also is this algorithm robust?**). (**Ram: I don't understand the relevance of these next to sentences...**) If configured properly[1], MSNs are expected to handle large region environmental mapping task efficiently while maintaining minimal overlapping sensing regions.

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 [9]. 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 [10]. 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], [4]. 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 [11], [12] . 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 [13]. 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.

[1]One popular example of this is the work of [8] where centroidal Voronoi configuration of MSNs are proven to achieves optimal coverage.

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, and reviews a recursive Bayesian filter tailored to the problem. Section III presents our combined probabilistic sensor model. 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 and proposes a number of future directions.

II. Problem Description

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

A. Notation

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 $\mathbb{Z}^+$ 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 $d = 2$ or 3. To simplify the presentation, we focus on the case when $d = 2$ although the framework generalizes to $d = 3$ in a straightforward manner (**Ram: is this right?**). Let $\mathbb{S}^{d-1} = \{s \in \mathbb{R}^d \mid \|s\| = 1\}$ be a unit sphere.

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 $x_t^i \in \mathcal{Q} \times \mathbb{S}^{d-1}$. We denote by the set $x_{0:t} := \{x_0, \ldots, x_t\}$ the robot states up to time t. Given a pair of states (x_t, x_{t+1}), robots follow a way-point-based, continuous-time, deterministic motion model under the following dynamics:

$$\dot{x}(l) = \mathrm{f}(x(l),\, u(l)),\;\; l \in [l_0,\, l_f] \tag{1}$$

with boundary conditions $x(l_0) = x_t$ and $x(l_f) = x_{t+1}$ where $u : [l_0, l_f] \to \mathbb{R}^k$ is a control input, l_0 is the *initial time*, and l_f is the *final time* which is free. Let $u_t^\star$ be an *optimal control policy* that drives the robots' state from x_t

to x_{t+1} in minimum time under the dynamics [2]. 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 that is 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 (quantitative information about the target), $\boldsymbol{I} \in \mathcal{I} \subseteq \mathbb{R}$. The Cartesian product $\mathcal{Z} = \mathcal{Q} \times \mathcal{I}$ is referred to as the *target state space*. Let $\boldsymbol{y}_t = (\boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^n)$ be a binary random n-tuple that indicates whether an observation was made by m robots at time step t where each $\boldsymbol{y}_t^i \in \{0, 1\}$, and n is the dimension of the sensor input. 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 $\mathcal{F} \subseteq \{1, \ldots, m\}$ be the index set of robots whose sensor has failed.

B. An illustrative Example: Airborne LiDAR for DEM generation

Consider a group of autonomous aerial vehicles trying to acquire an accurate Digital Elevation Model (DEM)[3] 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 measurement from some part of the fleet is corrupt or unreliable which could degrade the quality of DEM. Taking such adversarial scenarios into account, it is crucial to design a deployment strategy along given a sensor model capable of fault-detection which guarantees 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 tat within a reasonable time, the autonomous fleet could obtain an accurate DEM model.

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

For a given target located at $q \in \mathcal{Q}$, $p(\boldsymbol{y}_D^i = 0 \mid x^i, \boldsymbol{q} = q)$ is the probability that ith robot detected target located at q, and $\boldsymbol{y}_D^i = 1$ otherwise. In a similar manner, the joint event $p(\boldsymbol{y}_D = \boldsymbol{0} \mid x, \boldsymbol{q} = q)$ is the probability that m robots failed to detect the target at q. The optimal configuration—for the case when sensors in $\mathcal{F}$ fail to operate—must satisfy:

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

subject to

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

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[4]. 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*[5]. 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 [REFs] for robot target assignment, and will show that the solution given the assignment solves (2).

D. 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[6] *a priori*. Furthermore we 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 as a correct measurement. In other words, only the sensor measurement provided from a positive detection is considered, but others will be abandoned. These assumptions turned out to be unrestrictive according to the work of [13] in which authors has experimentally verified a similar combined sensor model on mobile robot using laser range finder and a panoramic camera measurement.

E. Environmental mapping using recursive Bayesian filter

Based on our sensor model, and the deployment strategy, we derive Bayesian filtering equation to recursively update beliefs on a particular unknown environment. Let b_t represent a the posterior probability distribution of the target state at time $t \in \mathbb{Z}_{\geq 0}$, the initial belief b_0 is assumed to be uniform density if no prior information on the target is available, and let $b^\star$ be the *true posterior belief*[7] To this end, we quantify the difference between the true posterior belief, $b^\star$ and an approximation using our method via the Kullback-Leibler (K-L) divergence. We demonstrate via our numerical simulation 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 $D_{\text{KL}}\left(b_t \| b^\star\right) < \epsilon$.

[2] In the current context, the optimal control policy is a sequence of control inputs (in discrete time domain) or control path (in continuous time domain) governed by dynamics of the vehicle. Though the focus of this paper is not generating such a policy, one could generate such a controller using a Linear-Quadratic Regulator (LQR) if the dynamics were linear.

[3] A digital elevation model (DEM) is a digital model or 3D representation of a terrain's surface commonly for a planet (including Earth), moon, or asteroid – created from terrain elevation data [THIS HAS BEEN TAKEN FROM WIKIPEDIA. I WILL NEED TO CHANGE THIS...].

[4] Worst-case optimal (minimax) solution to the problem could lead to too conservative in these type of applications.

[5] 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).

[6] Successful detection of a target means that the sensor can discern the target from others.

[7] 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.

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 is corrupted by noise, and the measurement is valid only if the target is detected. This combined sensor model joins the generic noisy sensor model with the binary detection model [11], [12]. In fact, this combined sensor model has been experimentally validated during an object mapping and detection task using a laser scanner [13]. 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 for a given $\mathcal{F}$, the probability of the event that all m sensors with configuration x_t fail to detect the target located at $q \in \mathcal{Q}$ is

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

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,

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

or (4) can be obtained via other distributed sensor fusion techniques (see e.g., [14]). 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 $\boldsymbol{y}_D = (\boldsymbol{y}_D^1, \ldots, \boldsymbol{y}_D^m)$ where $\boldsymbol{y}_D^i \in \{0, 1\}$ for all i when conditioned on $x_t, \boldsymbol{z}$, so that the conditional PDF can be computed as:

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

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

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

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

$$\begin{aligned} &f_{\boldsymbol{y}_t|\boldsymbol{z},x_t,\mathcal{F}}(\boldsymbol{y}_t = \mathbf{1} \mid z, x_t, \mathcal{F}) \\ &= (1 - \underbrace{p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t,\mathcal{F}}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid z, x_t, \mathcal{F}))}_{(\star)\text{ the probability of missded detection}} \\ &\quad \times \underbrace{f_{\boldsymbol{y}_t|\boldsymbol{z},x_t\boldsymbol{y}_{D,t},\mathcal{F}}(y_t \mid z, x_t, \boldsymbol{y}_{D,t} \neq \mathbf{0}, \mathcal{F})}_{\text{the likelihood of reliable measurements}}, \\ &+ I_{\text{range}}^{-1} p_{\boldsymbol{y}_{D,t}|\boldsymbol{z},x_t,\mathcal{F}}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid z, x_t, \mathcal{F}) \end{aligned} \tag{5}$$

By minimizing $(\star)$, one can ensure that only reliable measurements on the target state is considered.

IV. Deployment strategy

This section presents our class of robust deployment strategies. At each time, m robots move to new locations so as to minimize the missed-detection probability under sensor failures to promote one-step future observations. Since the set of robots with faulty sensors is unknown, we cannot solve to our problem for arbitrary number of sensor failures. Instead by setting a robot-target assignment rule based on space partitioning technique, we could however, ensure that every target is being detected by at least one robot, and such assignment is optimal in terms of minimizing Euclidean distance between robots and every target in $\mathcal{Q}$. This so called partitioned-based deployment is common to multi-robot coverage problems [8], [10], [15], [16]. The most popular one is based on the Voronoi tessellations (see e.g., [8], which we call a *non-robust deployment*). There are, in fact more general methods, which partition the workspace into p regions and assign $k \in \{1, \ldots, m\}$ robots each region (note that if $k = m$, the method becomes *centralized*) [10]. 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. Thus, if each sensor has an effective sensing range long enough to cover the whole workspace, utilizing all m sensors to detect every target in the workspace becomes the most desirable strategy.

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 [17], 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[8], 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:

Definition 4.1 (An Order−k Voronoi Partition [9]): 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:

$$\int_{\mathcal{Q}} p_{\boldsymbol{y}_D|x,\boldsymbol{q}}\left(\boldsymbol{y}_D = \mathbf{0} \mid x, \boldsymbol{q} = q\right) \phi(q)\, dq. \tag{6}$$

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

$$H(x, W, {}^kG) = \sum_{j=1}^{l} \int_{W^j} \left(\prod_{x^i \in {}^kG(W^j)} \left(1 - h\left(\|q - x^i\|\right)\right) \right) \phi(q)\, dq \tag{7}$$

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 ([16]): 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 $\mathcal{Q}$.

[8]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.

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:

$$p_{\boldsymbol{y}^i_{D,t}|\boldsymbol{q},x_{t-1},u_t}\left(\boldsymbol{y}^i_{D,t} = 1 \mid, x_{t-1}, u_t, \boldsymbol{q} = q\right) = \begin{cases} h\left(\left\|q - x_t^i\right\|\right) & \text{if } q \in {}^kG_t^\star(x_t^i) \cap \mathcal{B}(x_t^i, r_{\text{eff}}), \\ 0, & \text{otherwise,} \end{cases}$$

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 prior belief and the observations, robots compute/move their next way-points, and the posterior belief is updated at the new locations given the collected information. By using $f_{\boldsymbol{q}}$, and (7), for a given x_{t-1} and $\mathcal{F}$ at time $t-1$, we want to obtain the next way-point $x_t^\star$ which solves

$$x_t^\star \leftarrow \arg\min_{x_t} \Bigg\{ \mathcal{L}(x_t) := \sum_{j=1}^{l} \int_{W_t^j} \prod_{x_t^i \in {}^kG(W_t^j)} \left(1 - h\left(\left\|q - x_t^i\right\|\right)\right) f_{\boldsymbol{q}}(q)\, dq \Bigg\}. \tag{8}$$

Note that for a given k, by substituting $f_{\boldsymbol{q}}$ with ϕ and x with x_t, (8) takes the identical form as $H(x, W, G^k)$ which was previously defined in (7). 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,

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

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 via Invariance Principle. The theorem along with the formal proof is contained in our previous paper [16].

V. Implementation: environmental mapping

In this section, we first introduce Bayesian filtering equations under our particular target distribution setting, 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

Algorithm 1: Gradient Algorithm (MMLE)

Input: $\mathcal{L}_k$, f, $\hat{x}_{t-1}$, $\epsilon > 0$
Output: $\hat{u}_t$
$k \leftarrow 0$, $\Delta \leftarrow \epsilon$
while $\Delta > \epsilon$ **do**
  foreach $i \in \{1, \dots, m\}$ **do**
    $x^i_{t,k+1} \leftarrow x^i_{t,k} - \alpha^i_{t,k} \nabla_i \mathcal{L}_k(x_{t,k})$
    // $\alpha^i_{t,k}$ is obtained using a line search method
  $\Delta \leftarrow \mathcal{L}_k(x_{t,k}) - \mathcal{L}_k(x_{t,k+1})$
  $k \leftarrow k + 1$
$\hat{x}_t \leftarrow x_{t,k}$
$\hat{u}_t \leftarrow$ LQR($\hat{x}_t$, $\hat{x}_{t-1}$, f)
return $\hat{u}_t$

goal: environmental mapping by m robots. Recall that $b_t(z)$ represent a *belief* on target {information, location} state at time $t \in \mathbb{Z}_{\geq 0}$, the posterior probability distribution of the target state described by a random vector $\boldsymbol{z} \in \mathcal{Z}$. Thus, b_t describes the environmental map of some bounded region at time t. The state of robots are assumed completely *known*. In a similar manner, the belief of target information state $\boldsymbol{I}$ given the target located at q is given by:

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

where we denote the initial belief on target state by b_0. Note that $b_t(I \mid \boldsymbol{q} = q)$ depends on the initial belief b_0, the previous robot trajectories up until time t where $t \in \mathbb{Z}^+$, and observations up to this point, $\boldsymbol{y}_{1:t}$.

If the probability distribution about the target location, namely $f_{\boldsymbol{q}}$ is known *a priori*, the belief on the complete target state $\boldsymbol{z}$ is:

$$b_t(z) = f_{\boldsymbol{z}|b_0, x_{0:t}, \boldsymbol{y}_{1:t}}\left(z \mid b_0, x_{0:t}, y_{1:t}\right) = b_t(I \mid \boldsymbol{q} = q) f_{\boldsymbol{q}}(q). \quad (10)$$

If there is no prior knowledge of the target information at the initial time, one can choose the prior distribution as the *uniform* density. 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*, (9) becomes:

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

where $t \in \mathbb{N}$. One can simplify the likelihood function in the target information map by using this observation, which yields:

$$b_t(I \mid \boldsymbol{q} = q) = \eta_t \, f_{\boldsymbol{y}_t|\boldsymbol{z}, x_t}\left(y_t \mid \boldsymbol{z} = (I, q), x_t\right) b_{t-1}(I \mid \boldsymbol{q} = q) \quad (11)$$

$\eta_t := \left(f_{\boldsymbol{y}_t|\boldsymbol{q}, b_0, x_t}\left(y_t \mid \boldsymbol{q} = q, b_0, x_t\right)\right)^{-1}$ denotes the marginal probability, which is known as the *normalization constant*. This usually cannot be directly computed, but can be obtained by utilizing the total law of probability:

$$\eta_t = \left(\int_{\mathcal{I}} f_{\boldsymbol{y}_t|\boldsymbol{z}, x_t}\left(y_t \mid \boldsymbol{z} = (I, q), x_t\right) b_{t-1}(I \mid \boldsymbol{q} = q) \, dI \right)^{-1}$$

By joining the (10) and (11), one can obtain a simplified form of the filtering equation:

$$\begin{aligned} b_t(z) &= \eta_t \, f_{\boldsymbol{y}_t|\boldsymbol{z}, x_t}\left(y_t \mid z, x_t\right) b_{t-1}(z) \\ &= \left(\prod_{i=1}^{t} \eta_i f_{\boldsymbol{y}_i|\boldsymbol{z}, x_i}\left(y_i \mid z, x_i\right) \right) b_0(z). \end{aligned} \quad (12)$$

We assume that m robots share their beliefs.

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 [18]. We consider Sequential Importance Resampling (SIR) [19] 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.

$$q^1, \dots, q^{N_1} \quad (13)$$

where for each $i \in \{1, \dots, N_1\}$, $j \in \{1, \dots, N_2\}$,

$$\widetilde{w}_t^{ij} \propto f_{\boldsymbol{y}_t|\boldsymbol{z}_t, x_t}(\boldsymbol{y}_t - \mathbf{1} \mid \hat{x}_t, \, \boldsymbol{z} = (q^i, I^{ij})) \quad (14)$$

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

$$\begin{aligned} &\left\{ \left\{ \left(z^{11}, \widetilde{w}_t^{11}\right), \dots, \left(z^{1N_2}, \widetilde{w}_t^{1N_2}\right) \right\}, \right. \\ &\left\{ \left(z^{21}, \widetilde{w}_t^{21}\right), \dots, \left(z^{2N_2}, \widetilde{w}_t^{2N_2}\right) \right\}, \dots, \\ &\left. \left\{ \left(z^{N_1 1}, \widetilde{w}_t^{N_1 1}\right), \dots, \left(z^{N_1 N_2}, \widetilde{w}_t^{N_1 N_2}\right) \right\} \right\} \end{aligned}$$

where for each $i = 1, \dots, N_1$, $\sum_{j=1}^{N_2} \widetilde{w}_t^{ij} = 1$.

After resampling and normalizing, the approximate posterior belief becomes

$$\hat{b}_t(z) = \sum_{k=1}^{N} w_t^k \delta(z - z^k) \quad (15)$$

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 [20], 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 a class of deployment strategies. 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: $\hat{b}_{t-1} = \{z^l,\, w^l_{t-1}\}_{l=1}^N, y_t, \hat{x}_{t-1},\, I_{\text{range}}$
Output: $\hat{b}_t$
// Propogate motion model; see Algorithm 1
$\hat{u}_t \leftarrow \text{MMLE}(y_{D,t},\, \hat{x}_{t-1})$
// SIR Particle Filter
// 1) Update using the observation model
foreach $i \in \{1, \ldots, N_1\}$ **do**
  foreach $j \in \{1, \ldots, N_2\}$ **do**
    $\widetilde{w}^{ij}_t \leftarrow p_{\boldsymbol{y}_{D,t}|\boldsymbol{q},\hat{x}_{t-1},\hat{u}_t}(\boldsymbol{y}_{D,t} = \mathbf{0} \mid \boldsymbol{q} = q^i, \hat{x}_{t-1}, \hat{u}_t)(I^{-1}_{\text{range}} - w^{ij}_{t-1} f_{\boldsymbol{y}_t|\boldsymbol{z},\hat{x}_{t-1},\hat{u}_t,\boldsymbol{y}_{D,t}}(y_t \mid \boldsymbol{z} = z^{ij}_t, \hat{x}_{t-1}, \hat{u}_t, \boldsymbol{y}_{D,t} \neq \mathbf{0})) + w^{ij}_{t-1} f_{\boldsymbol{y}_t|\boldsymbol{z},\hat{x}_{t-1},\hat{u}_t,\boldsymbol{y}_{D,t}}(y \mid z, \hat{x}_{t-1}, \hat{u}_t, \boldsymbol{y}_{D,t} \neq \mathbf{0})$
// 2) Resample and Normalize
$\{w^l_t\}_{l=1}^N \leftarrow \text{Resample}(\{\widetilde{w}^l_t\}_{l=1}^N,\, \{w^l_{t-1}\}_{l=1}^N)$
return $\hat{b}_t \leftarrow \{z^l, w^l_t\}_{l=1}^N$
// Low Variance Resampling [21]
function Resample$(\{\widetilde{w}^l_t\}_{l=1}^N,\, \{w^l_{t-1}\}_{l=1}^N)$
  forall $i \in \{1, \ldots, N\}$ **do**
    $\overline{w}^i_t \leftarrow \frac{\widetilde{w}^i_t \cdot w^i_{t-1}}{\sum_{i=1}^N \widetilde{w}^i_t \cdot w^i_{t-1}}$
  foreach $i \in \{1, \ldots, N_1\}$ **do**
    $\delta \leftarrow \text{rand}((0; {N_2}^{-1}))$
    $cdf \leftarrow 0,\, k \leftarrow 0,\, c_j \leftarrow []$ for all j
    for $j = 0,\, j < N_2$ **do**
      $u \leftarrow \delta + j \cdot {N_2}^{-1}$
      while $u > cdf$ **do**
        $k \leftarrow k + 1$
        $cdf \leftarrow cdf + \overline{w}^{ik}_t$
      $c_{j+1} \leftarrow k$
    for $j = 1;\, j \leq N_2$ **do**
      $w^{ij}_t \leftarrow \frac{c_j}{N_2}$
  return $\hat{b}_t = \{z^l,\, w^l_t\}_{l=1}^N$

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, longitude. We let $\mathcal{I} = [-1000,\, 4000]$ range of elevation in feet, and $r_{\text{eff}} = 2.89$ miles (or ∞ if it is relaxed). Targets are uniformly distributed over $\mathcal{Q}$, and the initial expected value for initial target information over $\mathcal{Q}$ is given in Fig. 1 as a mixture of Gaussian kernels. The information vector ranges from -1000 to 4000, i.e., $I_{\min} = -1000$, $I_{\max} = 4000$. 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, nevertheless, we postulate that the results will be comparable due to the similarities of those idea.

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

Algorithm type:	gradient computation	related studies
non-robust	fully decentralized	[3], [8]
robust ($k = 2$)	decentralized	current paper
max. information gain	centralized	[1]–[5]

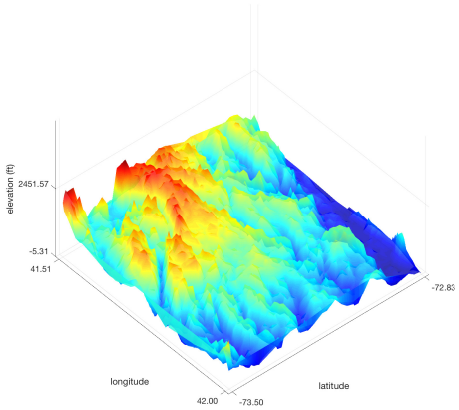

Fig. 1: Elevation map of some region in Connecticut (the ground truth).

A. Convergence of our deployment algorithm

First, the behavior of the deployment strategy is discussed. Given the initial uniform prior belief and the initial configuration at x_0, three algorithms, summarized in Table I, were tested. Fig. 2 shows the configurations after $t = 1$ with non-robust method (left) and robust method (right) where $k = 2$. Fig. 3 compares the convergence speed and the cost between three methods.

B. Environmental mapping/filtering performance w/o failure

Next, we present the evolution of the object map given the uniform, initial map (Fig. 1) with successive positive observations, each followed by the gradient descent strategy and filtering process. Fig. 4 illustrates the map building process, given limited effective sensing range value ($r_{\text{eff}} = 0.3$, the value was chosen empirically relative to the workspace size) when using our robust method. Others show similar performance, and are omitted due to the lack of space. Fig. 5 compares the K-L divergence values between different

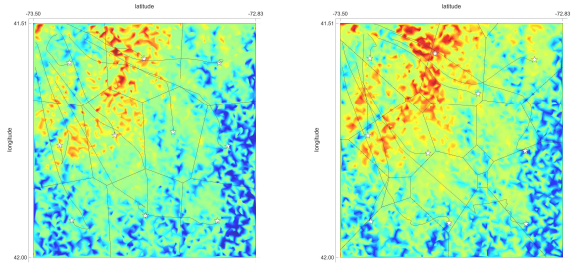

Fig. 2: One-time deployment with non-robust (left) and robust ($k = 2$) (right) overlaid with map at $t = 1$ (lines: gradient descent flow, stars: positions of robots, polygons: partition).

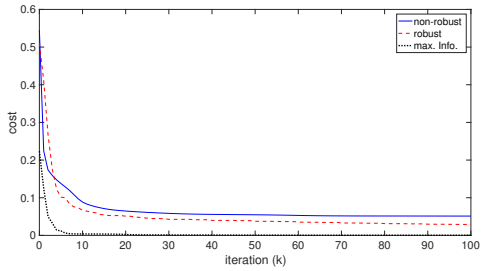

Fig. 3: Convergence test for one-time deployment with different methods.

strategies during the evolution. While the max. information gain approach shows the best result, other methods also shows competitive mapping performance compared to the ground truth.

C. Robustness to sensor failure

This section will present a number of examples when robust method become more appealing than non-robust method, which happens when the sensors are prone to fail. We consider three different cases, when $\mathcal{F} = \{1\}, \{1,2\}, \{1,2,3\}$. 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. 6. As can be seen from Fig 6 and Fig. 7, 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 6, both middle and right, the unmapped area is owe to the limited sensing range. It is not surprising to see from this example that the maximum information gain method shows the best robustness to sensor failures among the three; however this is due to the help from the central information fusion sever and relatively large communication load.

D. 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. 8 shows a distribution of K-L divergence values at the $t = 10$ for a given 100 random initial configurations with randomly sampled faults $1 \leq |\mathcal{F}| \leq 5$.

E. Scalability of our method

We also conducted a series of simulations and have validated that the previous results generalize to larger number of robots Fig. 3 shows one of the results with 50 robots after $t - 10$ where robots' sensing ranges (dashed lines) and the order-2 Voronoi partition (solid lines) are overlaid over the expected belief.

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 develop multi-agent patrolling algorithms (see e.g., [22]) to resolve such problems where there may not be enough sensors to cover the whole target space. Also, as reported in the literature [13], 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 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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

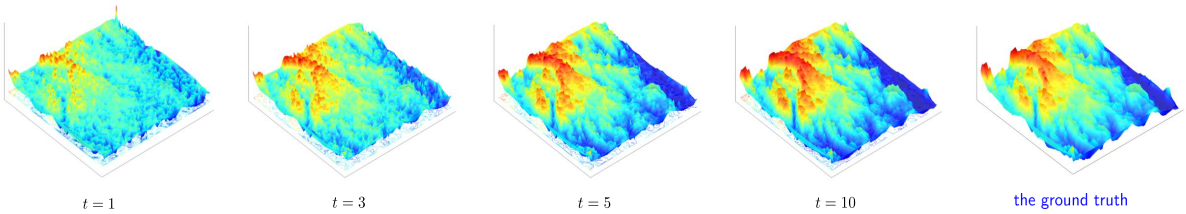

Fig. 4: Time evolution of expected belief with robust method ($k = 2$), $r_{\text{eff}} = 2.89$, $\Sigma_B = 0.04\mathbf{I}$.

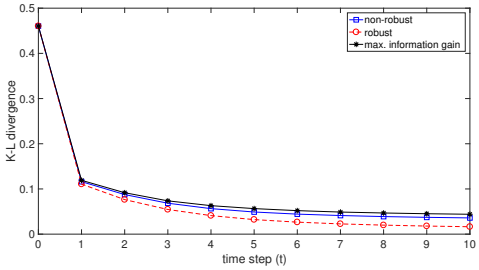

Fig. 5: Comparison of K-L divergence from the actual distribution between different methods during belief propagation.

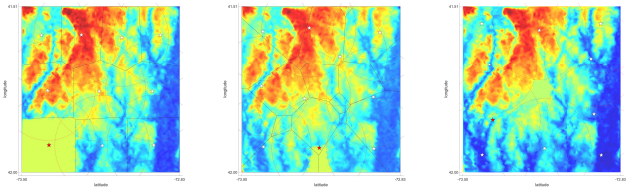

Fig. 6: The belief at $t = 10$ with non-robust method (left) and robust method (middle), and max. information method (right) Also showing robot configurations (stars, where red stars are robots whose sensors have failed), effective sensing range (dashed line), and partition (solid lines).

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

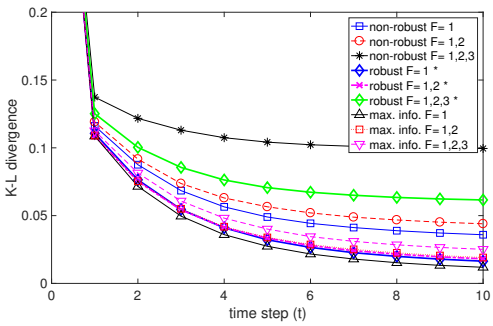

Fig. 7: Comparison of K-L divergence from the actual distribution between different methods during belief propagation when part of the sensor fail.

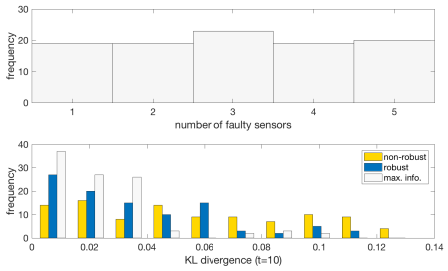

Fig. 8: Robustness test with 100 test dataset, distribution of $|\mathcal{F}|$ (top), K-L divergence at $t = 10$ (bottom).

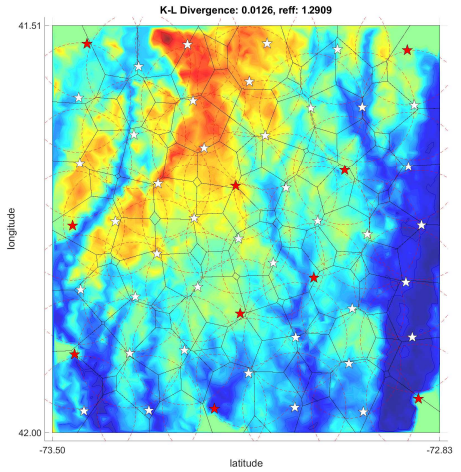

Fig. 9: Simulations with 50 robots where 20 sensors have failed.