A Multi-Robot Control Policy for Information Gathering in the Presence of Unknown Hazards

Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar

Abstract This paper addresses the problem of deploying a network of robots into an environment, where the environment is hazardous to the robots. This may mean that there are adversarial agents in the environment trying to disable the robots, or that some regions of the environment tend to make the robots fail, for example due to radiation, fire, adverse weather, or caustic chemicals. A probabilistic model of the environment is formulated, under which recursive Bayesian filters are used to estimate the environment events and hazards online. The robots must control their positions both to avoid sensor failures and to provide useful sensor information by following the analytical gradient of mutual information computed using these online estimates. Mutual information is shown to combine the competing incentives of avoiding failure and collecting informative measurements under a common objective. Simulations demonstrate the performance of the algorithm.

Mac Schwager
GRASP Lab, University of Pennsylvania, 3330 Walnut St, PA 19104, USA, and
Computer Science and Artificial Intelligence Lab, MIT, 32 Vassar St, Cambridge, MA 02139, USA
e-mail: `schwager@seas.upenn.edu`

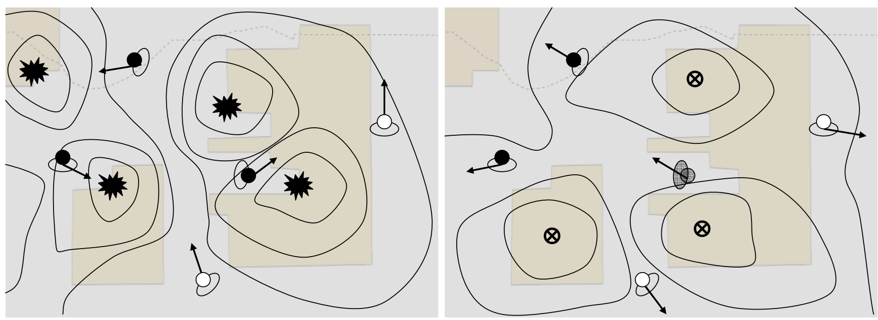
Philip Dames
GRASP Lab, University of Pennsylvania, 3330 Walnut St, PA 19104, USA
e-mail: `pdames@seas.upenn.edu`

Daniela Rus
Computer Science and Artificial Intelligence Lab, MIT, 32 Vassar St, Cambridge, MA 02139, USA
e-mail: `rus@csail.mit.edu`

Vijay Kumar
GRASP Lab, University of Pennsylvania, 3330 Walnut St, PA 19104, USA
e-mail: `kumar@seas.upenn.edu`

1 Introduction

Networks of robotic sensors have the potential to safely collect data over large-scale, unknown environments. They can be especially useful in situations where the environment is unsafe for humans to explore. In many such situations, robots are also susceptible to hazards. It is important to design exploration and mapping algorithms that are hazard-aware, so that the robotic sensor network can effectively carry out its task while minimizing the impact of individual robot failures. In this paper we propose an algorithm, based on an analytic expression for the gradient of mutual information, that enables a robotic sensor network to estimate a map of events in the environment while avoiding failures due to unknown hazards.



(a) Event map (b) Hazard map

Fig. 1 The tragic accident at the Fukushima nuclear power plant in Japan is a fitting scenario for our algorithm. Hypothetical maps of the events and hazards are shown over an image of the Fukushima plant from `http://maps.google.com/`. On the left, the events of interest are structural defects represented by the explosion symbols, and the contour lines represent the probability of detecting these defects. Sensors move to determine where the structural defects are by increasing the informativeness of their sensor readings. Black circles represent sensors that see a defect while white circles do not see a defect. On the right, the hazards are radiation sources represented by the $\otimes$ symbol, and the contours represent the probability of failure due to radiation damage. By moving to increase informativeness, the robots implicitly avoid hazards that may cause failure thereby preventing information from being collected. The grayed-out robot in the center has failed due to the high radiation levels.

Consider, for example, the recent tragic accident at the Fukushima nuclear power plant in Japan, which sustained critical damage from a large earthquake and tsunami in March, 2011. The algorithm we propose here could be used by a team of flying quadrotor robots with cameras to inspect the plant for structural damage, keeping human workers at a safe distance. With our algorithm the robots could build a map of the areas that are likely to have structural damage, while simultaneously building a map of the radiation hazards, to avoid failure due to radiation exposure. This scenario is illustrated in Fig. 1. Both the event map and the hazard map are estimated online using a recursive Bayesian filter, where the event map is estimated from evidence of structural damage seen by the cameras, and the hazard map is estimated by the previous failures of other robots. The robots move along the gradient of mutual information, which gives the direction of expected maximum information gain given the current maps, thereby driving the exploration of the environment. Our algorithm could also be used, for example, to monitor forest fires while avoiding fire damage, taking ocean measurements while avoiding damage from adverse weather, or mapping a chemical spill site while avoiding failure from caustic chemicals.

In all of these examples, the robots must move to both avoid hazards and provide useful sensor information. Although these two objectives may seem to be in conflict with one another, they are in fact complementary. If we want to map the events as precisely as possible, we implicitly want the robots to avoid hazardous areas, since the failure of a robot makes it unable to contribute to estimating the event map in the future. We use the gradient of mutual information to move the sensors so that their next measurements are as informative as possible. The gradient strategy blends the avoidance of hazards and the seeking of information into one probabilistically consistent objective. We propose a probabilistic model of the environment, the sensors, and the task, and derive the Bayesian filters for updating the event and hazard maps. We then prove a general theorem showing that the analytical gradient of mutual information has a simple form similar to mutual information itself. To our knowledge, this is the first proof of such an expression to appear in the literature. The mutual information gradient is then used to control the robots. We do not consider decentralization of the algorithm in this paper, though that will be a central concern of future work, and several existing methods can be adapted for decentralization.

1.1 Related Work

Mutual information is one of the fundamental quantities in information theory [Shannon, 1948, Cover and Thomas, 2006] and has been used extensively as a metric for robotic sensor network control and static sensor placement. For example, in [Grocholsky, 2002] and [Bourgault et al., 2002] mutual information is used as a metric for driving robotic sensor networks in gridded environments for target tracking and exploration tasks. Also, [Hoffmann and Tomlin, 2010] focused on decentralization and scalability using particle filters to approximate mutual information for target tracking. Recently in [Julian et al., 2011] the gradient of mutual information was used to drive a network of robots for general environment state estimation tasks, and a sampling method was employed to improve computational efficiency. The property of submodularity of mutual information was used in [Krause et al., 2006, Krause and Guestrin, 2007] for placing static sensors at provably near-optimal positions for information gain. The technique was extended to Gaussian processes in [Krause et al., 2008]. Approximations on information gain for static sensor placement were derived in [Choi et al., 2008, Choi and How, 2011b] and an informative trajectory planning algorithm was presented in [Choi and How, 2011a]. In a different but related application, [Vitus and Tomlin, 2010] uses mutual information to place static sensors to provide localization information to a mobile robot.

Our method differs in at least two important ways from those described above. Firstly, our work is specifically concerned with estimating and avoiding environmental hazards as well as estimating events. To our knowledge no works combining these objectives using mutual information have appeared in the literature. Secondly, we use an analytically derived expression for the gradient of mutual information for control. All the works above, save one, use grid-based finite difference methods to increase mutual information. The exception is [Julian et al., 2011], which employs the same analytical gradient of mutual information, but we have reserved the presentation of the proof of that result for this paper. A similar gradient result was derived in the context of channel coding in [Palomar and Verdú, 2007].

Many other methods that do not use mutual information have been proposed for mapping and exploring environments with robotic sensor networks. For example [Lynch et al., 2008] uses the error variance of a distributed Kalman filter to drive robots to estimate environmental fields. In [Schwager et al., 2009] a Voronoi based coverage algorithm from [Cortés et al., 2004] is augmented with online learning and exploration to estimate a scalar field in the environment. Similarly, [Martínez, 2010] expanded this coverage algorithm with online interpolation of an environmental field. Artificial potential fields have been used in [Howard et al., 2002] for multi-robot deployment and exploration, and[Li and Cassandras, 2005] uses a probabilistic coverage model for multi-robot deployment.

The question we address in this paper is: How do we choose the next positions $x_1, \dots, x_n$ to make the next Bayesian estimate of the event state as precise as possible? As already described, implicit in this question is the tendency to avoid hazards because a failed robot is an uninformative robot. However, in our scenario, as in real life, all the robots will eventually fail. To counteract the depletion of robots, we let there be a base station located in the environment that deploys new robots to replace ones that have failed. We let the rate of releasing new robots balance the rate of failed ones, so that the total number of robots is constant at all times. However many other interesting possibilities exist.

The rest of this paper is organized as follows. We define notation and formulate the problem in Section 2. We derive the Bayesian filters to estimate the hazards and events in Section 3. In Section 4 we derive the analytical gradient of mutual information and specialize it for our hazard-aware exploration problem. Finally, Section 5 presents the results of numerical simulations and conclusions and future work are discussed in Section 6.

2 Problem Formulation

Consider a situation in which n robots move in a planar environment $Q \subset \mathbb{R}^2$. The robots have positions $x_i(t) \in Q$ and we want to use them to sense the state of the environment while avoiding hazardous areas that may cause the robots to fail. Let the positions of all the robots be given by the vector $x = [x_1^T \cdots x_n^T]^T$. The robots give simple binary sensor measurements $y_i \in \{0, 1\}$ indicating wether or not they have sensed an event of interest near by. They also give a signal to indicate their failure status $f_i \in \{0, 1\}$, where $f_i = 1$ means that the robot has failed. Denote the random vector of all sensor outputs by $y = [y_1 \cdots y_n]^T$ and the vector of all failure statuses as $f = [f_1 \cdots f_n]^T$.

Table 1 List of symbols.

x_i	Position of sensor i	x	Stacked vector of sensor positions
y_i^t	Reading of sensor i at time t	$y^{1:t}$	Time history of measurements up to t
f_i^t	Failure status of sensor i at time t	$f^{1:t}$	Time history of failures up to t
q_j	Centroid position of grid cell j	Q	Environment
s_j	Event state in grid cell j	s	Full event state of environment
h_j	Hazard state in grid cell j	h	Full hazard state of environment

The task of the robot network is to estimate the state of the environment with as little uncertainty as possible. While the robots move in continuous space, we introduce a discretization of the environment to represent the environment state. We let the state of the environment be modeled by a random field $s = [s_1 \cdots s_{m_s}]^T$, in which each random variable s_j represents the value of the field at a position $q_j \in Q$ and m_s is the number of discrete locations. Each of these random variables takes on a value in a set $s_j \in S$, and the environment state has a value $s \in \mathscr{S} = S^{m_s}$. Similarly, the hazard level, which is related to the probability of failure of the robot, is modeled as a random field $h = [h_1^T \cdots h_{m_h}^T]$ in which h_k represents the hazard level at position $q_k \in Q$, and takes on a value in a set H. Then the hazard state of the whole environment has a value $h \in \mathscr{H} = H^{m_h}$. In general, S and H may be infinite sets, however one special case of interest is $S = \{0,1\}$, and $H = \{0,1\}$, so the state and hazard distributions denote the presence or absence of a target or a hazard, respectively, in a grid cell. Note that the use of the phrase grid cell refers to an element in the discretization of the environment, which need not be a square grid. We will work with the more general framework to include the possibility that some areas may be more important than others, or that there may be multiple events or hazards in a single grid cell. Also note that the discretization of the environment for state and hazard estimation need not be the same, for example we might need more precise localization of events than hazards. Let $\phi^0(s)$ and $\psi^0(h)$ denote the robots' initial guess at the distribution of the state and the hazards, respectively, which can be uniform if we have no prior information about the events or hazards.

Furthermore, in our scenario the robots have some probability of failure due to the hazards in the environment. Let the probability of failure of a robot at x_i due to hazard level h_j at location q_j be given by $\mathbb{P}(f_i = 1 \mid h_j = 1) = \alpha(x_i, q_j)$. We also assume that the hazards act independently of one another and that the probability of failure when infinitely far away from a hazard is given by $P_{\mathrm{f,far}}$, so that

$$\mathbb{P}(f_i = 0 \mid h) = (1 - P_{\mathrm{f,far}}) \prod_{j \mid h_j = 1} \mathbb{P}(f_i = 0 \mid h_j) = (1 - P_{\mathrm{f,far}}) \prod_{j \mid h_j = 1} \big(1 - \alpha(x_i, q_j)\big). \quad (1)$$

In words, the probability of a robot not failing is the product of the probability of it not failing due to any individual hazard. This gives the probability of a robot failing due to any number of hazards in the environment as

$$\mathbb{P}(f_i = 1 \mid h) = 1 - (1 - P_{\text{f,far}}) \prod_{j \mid h_j = 1} \big(1 - \alpha(x_i, q_j)\big). \tag{2}$$

When a robot fails, its sensor will output 0 with probability 1, that is, its sensor reading gives no indication of whether or not there is an event of interest near by, giving the conditional probability $\mathbb{P}(y_i = 1 \mid f_i = 1, s) = 0$. In this case, the sensor will naturally provide no further information about event or hazard locations.

If the robot does not fail, the sensor output, y_i, is a Bernoulli random variable with the probability of $y_i = 1$ due to a state value s_j at position q_j given by $\mathbb{P}(y_i = 1 \mid f_i = 0, s_j) = \mu(x_i, q_j)$, and the probability that $y_i = 0$ the complement of this. We again assume that state locations act independently on the robot's sensor and that the probability of a false positive reading is P_{fp} so that

$$\mathbb{P}(y_i = 0 \mid f_i = 0, s) = (1 - P_{\text{fp}}) \prod_{j \mid s_j = 1} \big(1 - \mu(x_i, q_j)\big). \tag{3}$$

Then the probability that a robot's sensor gives $y_i = 1$ for a given environment state is the complement of this,

$$\mathbb{P}(y_i = 1 \mid f_i = 0, s) = 1 - (1 - P_{\text{fp}}) \prod_{j \mid s_j = 1} \big(1 - \mu(x_i, q_j)\big). \tag{4}$$

We defer the discussion of specific forms of the functions α and μ to Section 5, however potential choices for the case where a hazard or event is present would be a decreasing exponential, a Gaussian function, or, in the simplest case, a constant (e.g. close to 1) inside some distance to q_j and some other constant (e.g. close to zero) outside. We will see that when μ or α have compact support in Q, there are computational benefits.

Now we consider the group of robots together. We derive three quantities that will be used in the Bayesian filters and control law; the likelihood function of the sensor measurements given the failures, the events, and the hazards, $\mathbb{P}(y \mid f, s, h)$; the likelihood function of the failures given the events and the hazards, $\mathbb{P}(f \mid s, h)$; and the likelihood function of the sensor measurements given the events and the hazards, $\mathbb{P}(y \mid s, h)$.

For $\mathbb{P}(y \mid f, s, h)$, assume that each robot's measurement is conditionally independent of the hazards and the other robots' measurements given its own failure status and the environment state, $\mathbb{P}(y \mid f, s, h) = \prod_{i=1}^{n} \mathbb{P}(y_i \mid f_i, s)$. Supposing we know the failure status of each robot, we can compute the measurement likelihood to be

$$\mathbb{P}(y \mid f, s, h) = \prod_{i \mid f_i = 0} \mathbb{P}(y_i \mid f_i = 0, s) \prod_{j \mid f_j = 1} \mathbb{P}(y_j \mid f_j = 1, s),$$

but

$$\prod_{j \mid f_j = 1} \mathbb{P}(y_j \mid f_j = 1, s) = \prod_{j \mid y_j = 0, f_j = 1} \mathbb{P}(y_j \mid f_j = 1, s) \prod_{j \mid y_j = 1, f_j = 1} \mathbb{P}(y_j \mid f_j = 1, s),$$

and the set $\{j \mid y=1, f_j=1\} = \emptyset$ and $\mathbb{P}(y_j = 0 \mid f_j = 1, s) = 1$, therefore this product reduces to 1. Then we have

$$\begin{aligned}\mathbb{P}(y \mid f,s,h) &= \prod_{i|y_i=0,f_i=0} \mathbb{P}(y_i = 0 \mid f_i = 0, s) \prod_{j|y_j=1,f_j=0} \mathbb{P}(y_j = 1 \mid f_j = 0, s) \times 1 \\ &= \prod_{i|y_i=0,f_i=0} \left(\prod_{k=1}^{m_s} \big(1-\mu(x_i,q_k)\big) \right) \prod_{j|y_j=1,f_j=0} \left(1 - \prod_{l=1}^{m_s} \big(1-\mu(x_j,q_l)\big) \right) \quad (5)\end{aligned}$$

where we use (4) and (3) to get the last equality.

Next we derive the failure likelihood, $\mathbb{P}(f \mid s,h)$. Conditioned on knowledge of the hazards, the failures are independent of the events and each other, so $\mathbb{P}(f \mid s,h) = \prod_{i=1}^{n} \mathbb{P}(f_i \mid h)$. Then using (1) and (2) we obtain

$$\mathbb{P}(f \mid s,h) = \prod_{i|f_i=0} \prod_{k=1}^{m_h} \big(1-\alpha(x_i,q_k)\big) \prod_{j|f_j=1} \Big(1 - \prod_{l=1}^{m_h} \big(1-\alpha(x_j,q_l)\big)\Big). \quad (6)$$

Finally, in the case that we want to predict an information gain for a future measurement, the failures are not yet known, so we will need the quantity $\mathbb{P}(y \mid s,h)$. This leads to

$$\begin{aligned}\mathbb{P}(y \mid s,h) &= \sum_{f\in\{0,1\}^n} \mathbb{P}(y \mid f,s,h)\mathbb{P}(f \mid s,h) = \sum_{f\in\{0,1\}^n} \prod_{i=1}^{n} \mathbb{P}(y_i \mid f_i,s)\mathbb{P}(f_i \mid h) \\ &= \prod_{i=1}^{n} \sum_{f_i\in\{0,1\}} \mathbb{P}(y_i \mid f_i,s)\mathbb{P}(f_i \mid h), \quad (7)\end{aligned}$$

where, as we already saw, $\mathbb{P}(f_i = 0 \mid h) = \prod_{j=1}^{m_h}(1-\alpha(x_i,q_j))$ and $\mathbb{P}(f_i = 1 \mid h) = 1 - \mathbb{P}(f_i = 0 \mid h)$, and similarly $\mathbb{P}(y_i = 0 \mid f_i = 0, s) = \prod_{j=1}^{m_s}(1-\mu(x_i,q_j))$ and $\mathbb{P}(y_i = 1 \mid f_i = 0, s) = 1 - \mathbb{P}(y_i = 0 \mid f_i, s)$, and finally $\mathbb{P}(y_i = 0 \mid f_i = 1, s) = 1$ and $\mathbb{P}(y_i = 1 \mid f_i = 1, s) = 0$. Next we use these quantities to derive a recursive Bayesian filters for maintaining a distribution over all possible event and hazard states.

3 Bayesian Estimation

As our robots move about in the environment, we wish to make use of their measurements and failure statuses at each time step in order to recursively estimate the events and hazards in the environment. We will show in this section, surprisingly, that the event and hazard estimates are either statistically independent, or they are deterministically linked (knowledge of either one fully determines the other). An unexpected consequence of the natural formulation in Section 2 is that there can be no statistical dependence between the event and hazard estimates except these two extremes. We let the robots collect measurements y^t synchronously at times $t = 1, 2, \ldots$, and we denote the tuple of all measurements up to time t by $y^{1:t} := (y^1, \ldots, y^t)$. We use a similar notation for failures, so that $f^{1:t} := (f^1, \ldots, f^t)$ is the tuple of all failure statuses up to time t. Furthermore, define the event distribution estimate up to time t to be $\phi^t(s) := \mathbb{P}(s \mid y^{1:t}, f^{1:t})$ and the hazard distribution up to time t to be $\psi^t(h) := \mathbb{P}(h \mid y^{1:t}, f^{1:t})$. The main result of this section is stated in the following theorem.

Theorem 1 (Bayesian Filtering). *The distributions for hazards and events given all information up to time t are independent with* $\mathbb{P}(s,h \mid y^{1:t}, f^{1:t}) = \phi^t(s)\psi^t(h)$*, assuming that h and s are not deterministically linked, and that their initial distributions are independent,* $\mathbb{P}(s,h) = \phi^0(s)\psi^0(h)$*. Furthermore,* $\phi^t(s)$ *and* $\psi^t(h)$ *can be computed recursively with the Bayesian filters*

$$\phi^t(s) = \frac{\mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)}{\sum_{s\in\mathscr{S}} \mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)}, \tag{8}$$

and

$$\psi^t(h) = \frac{\mathbb{P}(f^t \mid h)\psi^{t-1}(h)}{\sum_{h\in\mathscr{H}} \mathbb{P}(f^t \mid h)\psi^{t-1}(h)}. \tag{9}$$

In the case that the events and the hazards are deterministically linked, the Bayesian filter update for the distribution is given by

$$\mathbb{P}(s \mid y^{1:t}, f^{1:t}) = \frac{\mathbb{P}(y^t \mid f^t, s)\mathbb{P}(f^t \mid s)\mathbb{P}(s \mid y^{1:t-1}, f^{1:t-1})}{\sum_{s\in\mathscr{S}} \mathbb{P}(y^t \mid f^t, s)\mathbb{P}(f^t \mid s)\mathbb{P}(s \mid y^{1:t-1}, f^{1:t-1})}. \tag{10}$$

Proof. We will argue the existence of these two distinct cases by mathematical induction. We first prove (9) and then use it to prove (8). We will then derive (10) directly from the assumption that s and h are deterministically related.

To obtain an inductive argument, suppose that at $t-1$ the hazard estimate $\psi^{t-1}(h) = \mathbb{P}(h \mid y^{1:t-1}, f^{1:t-1}) = \mathbb{P}(h \mid f^{1:t-1})$ is independent of the sensor measurements $y^{1:t-1}$. Then the recursive Bayesian filter update for time t gives

$$\psi^t(h) = \frac{\mathbb{P}(y^t, f^t \mid h)\psi^{t-1}(h)}{\sum_{h\in\mathscr{H}} \mathbb{P}(y^t, f^t \mid h)\psi^{t-1}(h)}.$$

Now assuming that h and s are not deterministically related, we get $\mathbb{P}(y^t, f^t \mid h) = \mathbb{P}(y^t \mid f^t, h)\mathbb{P}(f^t \mid h) = \mathbb{P}(y^t \mid f^t)\mathbb{P}(f^t \mid h)$, where the last equality is because given the failure, f^t, the measurement, y^t, is independent of the hazards, h, as described in the previous section. This leads to

$$\psi^t(h) = \frac{\mathbb{P}(y^t \mid f^t)\mathbb{P}(f^t \mid h)\psi^{t-1}(h)}{\mathbb{P}(y^t \mid f^t)\sum_{h\in\mathscr{H}} \mathbb{P}(f^t \mid h)\psi^{t-1}(h)},$$

and we can cancel the factor of $\mathbb{P}(y^t \mid f^t)$ from the numerator and denominator to obtain (9). Now notice that $\psi^t(h) = \mathbb{P}(h \mid y^{1:t}, f^{1:t}) = \mathbb{P}(h \mid f^{1:t})$ remains independent of the measurements at time t. The initial distribution, $\psi^0(h)$, must be independent of $y^{1:t}$ (because no measurements have been collected yet), therefore by mathematical induction the hazard estimate distribution conditioned on the failures is always independent of the measurements.

Using a similar mathematical induction argument, suppose that the hazard and event estimates are independent given the measurements and failures up to time $t-1$, so $\mathbb{P}(h,s \mid y^{1:t-1}, f^{1:t-1}) = \phi^{t-1}(s)\psi^{t-1}(h)$. Then the Bayesian update for their joint distribution at time t is given by

$$\mathbb{P}(s,h \mid y^{1:t}, f^{1:t}) = \frac{\mathbb{P}(y^t, f^t \mid s,h)\phi^{t-1}(s)\psi^{t-1}(h)}{\sum_{s\in\mathscr{S}}\sum_{h\in\mathscr{H}}\mathbb{P}(y^t, f^t \mid s,h)\phi^{t-1}(s)\psi^{t-1}(h)}.$$

Factoring the numerator using the conditional independences described in Section 2, we get

$$\mathbb{P}(s,h \mid y^{1:t}, f^{1:t}) = \frac{\mathbb{P}(y^t \mid f^t, s)\mathbb{P}(f^t \mid h)\phi^{t-1}(s)\psi^{t-1}(h)}{\sum_{s\in\mathscr{S}}\sum_{h\in\mathscr{H}}\mathbb{P}(y^t \mid f^t, s)\mathbb{P}(f^t \mid h)\phi^{t-1}(s)\psi^{t-1}(h)},$$

and separating terms that depend on s from those that depend on h yields

$$\mathbb{P}(s,h \mid y^{1:t}, f^{1:t}) = \frac{\mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)}{\sum_{s\in\mathscr{S}}\mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)} \frac{\mathbb{P}(f^t \mid h)\psi^{t-1}(h)}{\sum_{h\in\mathscr{H}}\mathbb{P}(f^t \mid h)\psi^{t-1}(h)}.$$

We recognize the right most fraction as the Bayesian update from (9), and the left most expression can be factored as $\mathbb{P}(s,h \mid y^{1:t}, f^{1:t}) = \mathbb{P}(s \mid h, y^{1:t}, f^{1:t})\psi^t(h)$, which gives

$$\mathbb{P}(s \mid h, y^{1:t}, f^{1:t})\psi^t(h) = \frac{\mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)}{\sum_{s\in\mathscr{S}}\mathbb{P}(y^t \mid f^t, s)\phi^{t-1}(s)}\psi^t(h).$$

The fraction on the right is independent of h, so we conclude that $\mathbb{P}(s \mid h, y^{1:t}, f^{1:t}) = \mathbb{P}(s \mid y^{1:t}, f^{1:t}) = \phi^t(s)$, and we obtain the Bayesian update in (8). Therefore if the estimate distributions of s and h are independent at time $t-1$ they will also be so at time t, and by induction, if their initial distributions are independent then they will remain so for all time.

Finally, in the case that the hazards and the events are deterministically related, the standard recursive Bayesian filter yields

$$\mathbb{P}(s \mid y^{1:t}, f^{1:t}) = \frac{\mathbb{P}(y^t, f^t \mid s)\mathbb{P}(s \mid y^{1:t-1}, f^{1:t-1})}{\sum_{s\in\mathscr{S}}\mathbb{P}(y^t, f^t \mid s)\mathbb{P}(s \mid y^{1:t-1}, f^{1:t-1})},$$

which factors straightforwardly to the expression in (10). □

For the remainder of the paper, we consider the case where the hazards and the events are not deterministically related, so the filtering equations are given by (8) with (5) for the event update, and (9) with (6) for the hazard update. The robots use these filter equations to maintain estimates of the events and hazards in the environment. Next we consider using these estimates to derive an information seeking controller.

4 Control Using the Mutual Information Gradient

In this section we will derive an analytic expression for the gradient of mutual information in general terms, then specialize it to use the distributions of events and hazards found with the Bayesian filters in the previous section. This will lead to an information seeking controller for our robots.

In information theory [Cover and Thomas, 2006, Shannon, 1948], the mutual information between two random vectors s and y is defined as

$$I(S;Y) = \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \mathbb{P}(s,y)\log\frac{\mathbb{P}(s,y)}{\mathbb{P}(s)\mathbb{P}(y)}\,ds\,dy,$$

where we let e be the base of the logarithm, and $\mathscr{S}$ and $\mathscr{Y}$ are the range of s and y, respectively. We write s and y as though they were continuous valued for simplicity, though similar expressions can be written for discrete valued and general random variables. Mutual information indicates how much information one random variable gives about the other. In our scenario, we want to position our robots so that their next measurement gives the maximum amount of information about the event distribution.

Consider the situation in which the distribution $\mathbb{P}(s,y)$ depends on a parameter vector $x \in \mathbb{R}^{2n}$. We write $\mathbb{P}_x(s,y)$ to emphasize this dependence. Likewise, let $\mathbb{P}_x(s) := \int_{y\in\mathscr{Y}} \mathbb{P}_x(s,y)\,dy$, $\mathbb{P}_x(y) := \int_{s\in\mathscr{S}} \mathbb{P}_x(s,y)\,ds$, and

$$I_x(S;Y) := \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \mathbb{P}_x(s,y)\log\frac{\mathbb{P}_x(s,y)}{\mathbb{P}_x(s)\mathbb{P}_x(y)}\,ds\,dy. \tag{11}$$

In our case x is the positions of the robots, but the following result holds in a general context. We can compute the gradient of the mutual information with respect to the parameters x using the following theorem.

Theorem 2 (Mutual Information Gradient). *Let random vectors s and y be jointly distributed with distribution $\mathbb{P}_x(s,y)$ that is differentiable with respect to the parameter vector $x \in \mathbb{R}^{2n}$ over $Q^n \subset \mathbb{R}^{2n}$. Also, suppose that the support $\mathscr{S}\times\mathscr{Y}$ of $\mathbb{P}_x(s,y)$ does not depend on x. Then the gradient of the mutual information with respect to the parameters x over Q^n is given by*

$$\frac{\partial I_x(S;Y)}{\partial x} = \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}_x(s,y)}{\partial x}\log\frac{\mathbb{P}_x(s,y)}{\mathbb{P}_x(s)\mathbb{P}_x(y)}\,ds\,dy. \tag{12}$$

Proof. The theorem follows straightforwardly by applying the rules of differentiation. Notably, several terms cancel to yield the simple result. Differentiating (11) with respect to x, we can move the differentiation inside the integrals since $\mathscr{S}$ and $\mathscr{Y}$ do not depend on the parameters x. Then applying the chain rule to the integrand results in

$$\frac{\partial I(S;Y)}{\partial x} = \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}(s,y)}{\partial x} \log\frac{\mathbb{P}(s,y)}{\mathbb{P}(s)\mathbb{P}(y)}\,ds\,dy + \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \mathbb{P}(s,y)\frac{\mathbb{P}(s)\mathbb{P}(y)}{\mathbb{P}(s,y)}$$
$$\times\Big[\frac{\partial \mathbb{P}(s,y)}{\partial x}\frac{1}{\mathbb{P}(s)\mathbb{P}(y)} - \frac{\partial \mathbb{P}(s)}{\partial x}\frac{\mathbb{P}(y)\mathbb{P}(s,y)}{(\mathbb{P}(s)\mathbb{P}(y))^2} - \frac{\partial \mathbb{P}(y)}{\partial x}\frac{\mathbb{P}(s)\mathbb{P}(s,y)}{(\mathbb{P}(s)\mathbb{P}(y))^2}\Big]\,ds\,dy,$$

where we have suppressed the dependence on x to simplify notation. Bringing $1/(\mathbb{P}(s)\mathbb{P}(y))$ in front of the brackets gives

$$\frac{\partial I(S;Y)}{\partial x} = \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}(s,y)}{\partial x} \log\frac{\mathbb{P}(s,y)}{\mathbb{P}(s)\mathbb{P}(y)}\,ds\,dy$$
$$+\int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \Big[\frac{\partial \mathbb{P}(s,y)}{\partial x} - \frac{\partial \mathbb{P}(s)}{\partial x}\mathbb{P}(y\mid s) - \frac{\partial \mathbb{P}(y)}{\partial x}\mathbb{P}(s\mid y)\Big]\,ds\,dy.$$

Consider the three terms in the second double integral. We will show that each of these terms is identically zero to yield the result in the theorem. For the first term we have

$$\int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}(s,y)}{\partial x}\,ds\,dy = \frac{\partial}{\partial x}\int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \mathbb{P}(s,y)\,ds\,dy = \frac{\partial}{\partial x}1 = 0.$$

For the second term we have

$$\int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}(s)}{\partial x}\mathbb{P}(y\mid s)\,ds\,dy =$$
$$\int_{s\in\mathscr{S}} \frac{\partial \mathbb{P}(s)}{\partial x}\Big(\int_{y\in\mathscr{Y}} \mathbb{P}(y\mid s)\,dy\Big)\,ds = \frac{\partial}{\partial x}\int_{s\in\mathscr{S}} \mathbb{P}(s)\,ds = 0,$$

and the third term follows similarly if we interchange y and s. □

Remark 1. The result holds for the general definition of mutual information and makes no assumptions as to the distribution of the random variables, or the form of the dependence of $\mathbb{P}_x(s,y)$ on its parameters. The result also holds for generally distributed random variables including discrete valued ones (although we have written the theorem for continuous valued ones).

Remark 2. It is interesting that the gradient of $I_x(S;Y)$ has the same form as $I_x(S;Y)$ itself, except that the first occurrence of $\mathbb{P}_x(s,y)$ is replaced by its gradient with respect to x. To the authors' knowledge, this analytic expression for the gradient of mutual information has not been reported in the literature despite the proliferation of gradient based methods for maximizing mutual information in various applications ranging from channel coding [Palomar and Verdú, 2006, Palomar and Verdú, 2007], to medical imaging alignment [Viola and Wells III, 1997], to the control of robotic sensor networks [Grocholsky, 2002]. In [Palomar and Verdú, 2007] the authors derive a special case of Theorem 2 in which $\mathbb{P}(s)$ is not a function of x.

Remark 3. A similar expression holds for arbitrary derivatives of mutual information. For example, the Hessian of mutual information can be shown to be

$$\frac{\partial^2 I_x(S;Y)}{\partial x^2} = \int_{y\in\mathscr{Y}}\int_{s\in\mathscr{S}} \frac{\partial^2 \mathbb{P}_x(s,y)}{\partial x^2} \log \frac{\mathbb{P}_x(s,y)}{\mathbb{P}_x(s)\mathbb{P}_x(y)}\, ds\, dy$$

with essentially the same proof. In our robotic sensing scenario, this could be used to examine the coupling between the control laws for neighboring robots.

We will use the result in Theorem 2 to design a controller for our robotic sensor network. Writing the result in terms of quantities that we already know, we have

$$\frac{\partial I_x(S;Y)}{\partial x} = \sum_{y\in\{0,1\}^n}\sum_{s\in\mathscr{S}}\sum_{h\in\mathscr{H}} \frac{\partial \mathbb{P}_x(y \mid s,h)}{\partial x} \phi^t(s)\psi^t(h) \times \log \frac{\sum_{h\in\mathscr{H}} \mathbb{P}_x(y \mid s,h)\psi^t(h)}{\sum_{s\in\mathscr{S}}\sum_{h\in\mathscr{H}} \mathbb{P}_x(y \mid s,h)\phi^t(s)\psi^t(h)}, \quad (13)$$

where $\phi^t(s)$ and $\psi^t(h)$ come from the Bayesian filter equations (8) and (9), respectively, and $\mathbb{P}_x(y \mid s,h)$ comes from (7). The only remaining necessary term is the gradient of the measurement likelihood, which we can compute straightforwardly from (7) to be

$$\frac{\partial \mathbb{P}_x(y_i = 1 \mid s,h)}{\partial x} = \mathbb{P}(f_i = 0 \mid h)\mathbb{P}(y_i = 0 \mid s,h)\sum_{j|s_j=1} \frac{1}{1-\mu(x_i,q_j)} \frac{\partial \mu(x_i,q_j)}{\partial x_i} - \mathbb{P}(f_i = 0 \mid h)\mathbb{P}(y_i = 1 \mid s,h)\sum_{k|h_k=1} \frac{1}{1-\alpha(x_i,q_k)} \frac{\partial \alpha(x_i,q_k)}{\partial x_i}, \quad (14)$$

and when $y_i = 0$ it is simply the negative of this, $\frac{\partial \mathbb{P}_x(y_i=0|s,h)}{\partial x} = -\frac{\partial \mathbb{P}_x(y_i=1|s,h)}{\partial x}$. We propose to use an information seeking controller of the form

$$x_i(t+1) = x(t) + k\frac{\frac{\partial I_x(S;Y)}{\partial x_i}}{\left\|\frac{\partial I_x(S;Y)}{\partial x_i}\right\| + \varepsilon}, \quad (15)$$

where $k > 0$ is a maximum step size, and $\varepsilon > 0$ is a small factor to prevent singularities when a local minimum of mutual information is reached. Although this controller uses a gradient, it is not, strictly speaking, a gradient controller and a formal analysis of its behavior would be expected to be quite difficult. This is because the mutual information changes at each time step due to the integration of new sensor measurements into the Bayesian event estimate and hazard estimate. Intuitively, the controller moves the robots in the direction of the highest immediate expected information gain. An alternative approach would be to use a finite time horizon over which to plan informative paths using the information gradient. Indeed, our controller can be seen as a special case of this with a horizon length of one time step. Such an approach would have an exponential time complexity in the length of the horizon, however, making even a short time horizon computationally impractical.

Empirically, the controller drives the robots to uncertain areas while veering away from suspected hazard sites, learned through the failures of previous robots. The robots eventually come to a stop when they estimate the event state s of the environment with high confidence (i.e. when the entropy of $\phi^t(s)$ approaches zero). While hazard avoidance does not explicitly appear in the control law, it is implicit as an expected robot failure will decrease the amount of information gained at the next time step. This is true even when robots are replaced, since the mutual information gradient is agnostic to the replacement of sensors. It only considers the information gain it expects to achieve with its current robots by moving them in a particular direction. So when deciding in what direction to move, each robot balances the benefit of information received with the possibility of failure by moving in that direction. As future work, it would also be interesting to explicitly account for sensor losses by enforcing a total sensor loss budget, or a maximum sensor loss rate.

4.1 Computational Considerations

Unfortunately, one can see from (13) that the computation of $\partial I_x(S;Y)/\partial x$ is, in general, in $O(n2^n|\mathscr{H}||\mathscr{S}|)$ where n is the number of sensors. For example, if the hazard and event states are both binary, we have a complexity of $O(n2^{n+m_s+m_h})$, where m_s is the number of event grid cells and m_h is the number of hazard grid cells. Therefore this algorithm is not computationally practical for even a moderate number of robots or grid cells.

We are currently investigating two methods for decreasing the complexity of the control strategy, which will be detailed in a paper that is soon to follow. One involves a successive grid refinement procedure. In this procedure the robots begin with a coarse grid. Those grid cells that reach a threshold probability of containing an event or hazard are re-partitioned into a finer grid, while those grid cells with small probability are lumped together into a large cell. This procedure has the benefit of delivering arbitrarily fine event and hazard maps. The simulations described in Section 5 use a simplified version of this procedure. The second method that we are investigating is to assume a limited correlation structure among the grid cells in the environment. If two grid cells are statistically independent when they are separated by more than a fixed distance, the algorithm complexity reduces significantly. There also exist other methods for overcoming computational limitations, which include using Monte Carlo methods to approximate the sums in (13) in a decentralized way, as in [Julian et al., 2011], or particle filters as in [Hoffmann and Tomlin, 2010].

5 Simulation Results

We carried out Matlab simulations with the controller (15) over $\mathbb{R}^2$ with the sensor detection probability

$$\mu(x_i,q_j) = P_{\text{fp}} + \frac{1-P_{\text{fp}}-P_{\text{fn}}}{1+\exp\left(c(\|x_i-q_j\|-r_{\text{sense}})\right)},$$

with $P_{\text{fp}} = 0.01, P_{\text{fn}} = 0.05, c = 1, r_{\text{sense}} = 2$, and the robot failure probability

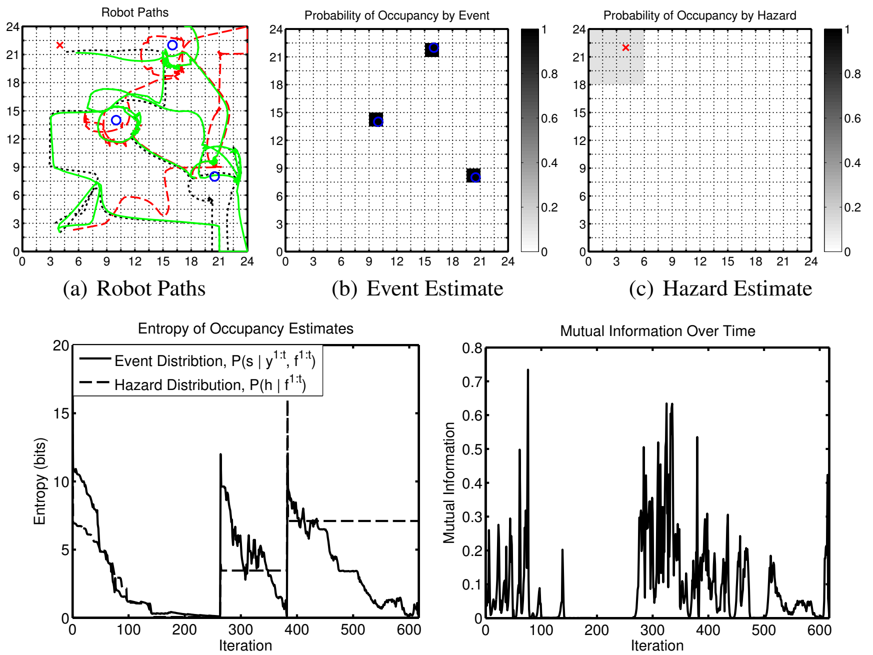
$$\alpha(x_i,q_j) = P_{\text{f,far}} + (1-P_{\text{f,far}}-P_{\text{s,near}})\exp\left(-\frac{\|x_i-q_j\|^2}{2\sigma_{\text{fail}}^2}\right),$$

with $P_{\mathrm{f,far}} = 0, P_{\mathrm{s,near}} = 0.1, \sigma_{\mathrm{fail}} = 1.25$, and uniform initial event and hazard distributions. The control gains used in (15) were $k = 0.1$ and $\varepsilon = 10^{-10}$. We used a simplified version of the grid refinement procedure described in Section 4.1 in which the environment was first divided into a 4×4 grid until the entropy of the event distribution dropped below 0.1, indicating a high degree of certainty in the estimates at that resolution. Then the entire grid was refined to 8×8, and when the entropy again dropped below 0.1, it was refined to 16×16. At any time if the probability of occupancy for an event (or hazard) was less than 10^{-15} for a cell, its occupancy probability was set to zero, the event (or hazard) distribution was renormalized appropriately, and that cell was ignored in all future event (or hazard) distribution updates. This successive grid refinement and pruning was found to dramatically improve computation speed.

The results of simulations with three robots, three events, and one hazard are shown in Fig. 2. The failure of two robots can be seen in Fig. 2(a) by the discontinuities in the green and black robot paths (new robots are introduced at the lower left corner of the environment to compensate for the failed ones). The event distribution is shown in Fig. 2(b) where the locations of the three events have been localized down to one grid square. Similarly, Fig. 2(c) shows the hazard distribution, in which the hazard has been localized to a 4×4 block of grid squares. The robots do not seek to refine the hazard estimate because it is sufficient for them to avoid the hazard an continue mapping the events. The decreasing trend in the entropy (or uncertainty) of the event and hazard estimates can be seen in Figs. 2(d). The mutual information, shown in Fig. 2(e), can be interpreted as the *expected* decrease in entropy, hence it tends to be large when there are large entropy drops. The entropy jumps at iteration 266 and 383, when the grid is refined to 8×8 and 16×16, respectively.

6 Conclusions

In this paper we proposed a multi-robot control policy that utilizes measurements about events of interest to locally increase the mutual information, while also using the history of robot failures to avoid hazardous areas. The central theoretical contribution of this paper is a new analytical expression for the gradient of mutual information presented in Theorem 2, which provides a principled approach to exploration by calculating robot trajectories that lead to the greatest immediate information gain. Despite minimal data from the binary sensors and a binary failure signal, the robot team is able to successfully localize events of interest and avoid hazardous areas. The event state and hazard fields over the environment are estimated using recursive Bayesian filters. The main drawback of the approach is high computational complexity, which makes the controller difficult to compute in realistic environments. We are currently investigating several techniques for reducing the complexity in order to make the algorithm more practically feasible. We are also investigating methods to decentralize the algorithm to run over a multi-robot network.



(a) Robot Paths (b) Event Estimate (c) Hazard Estimate

(d) Event and Hazard Estimate Entropy (e) Mutual Information

Fig. 2 This figure shows simulation results for three robots in an environment initially divided into a 4×4 grid, then refined to an 8×8 grid at the 266th iteration, and finally to a 16×16 grid at the 383rd iteration. Frame 2(a) shows the paths of the three robots which start from the lower left of the environment. The red '×' marks the location of a hazard and the three blue 'O's show the locations of events. The robots with the solid green and dotted black paths both fail once when they come too close to the hazard, after which two replacement robots are introduced at the lower left. Frames 2(b) and 2(c) show the final event and hazard distribution estimate, respectively. The events have been localized down to one grid square and the hazard down to a 4×4 region. Frame 2(d) shows the entropy, a measure of uncertainty, of the event (solid) and hazard (dotted) estimates. Both entropies decrease as the robots learn the event and hazard locations. They jump at the 266th and 383rd iteration when the grid is refined, as expected. The mutual information (or the expected drop in entropy) verses time is shown in frame 2(e).

Acknowledgements This work was funded in part by ONR MURI Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031, and the SMART Future Mobility project. We are grateful for this financial support.

References

[Bourgault et al., 2002] Bourgault, F., Makarenko, A. A., Williams, S. B., Grocholsky, B., and Durrant-Whyte, H. F. (2002). Information based adaptive robotic exploration. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS 02)*, pages 540–545.

[Choi and How, 2011a] Choi, H.-L. and How, J. P. (2011a). Continuous trajectory planning of mobile sensors for informative forecasting. *Automatica*. Accepted.

[Choi and How, 2011b] Choi, H.-L. and How, J. P. (2011b). Efficient targeting of sensor networks for large-scale systems. *IEEE Transactions on Control Systems Technology*. Submitted.

[Choi et al., 2008] Choi, H.-L., How, J. P., and Barton, P. I. (2008). An outer-approximation algorithm for generalized maximum entropy sampling. In *Proceedings of the American Control Conference*, pages 1818–1823, Seattle, WA, USA.

[Cortés et al., 2004] Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.

[Cover and Thomas, 2006] Cover, T. and Thomas, J. (2006). *Elements of Information Theory*. John Wiley and Sons, 2 edition.

[Grocholsky, 2002] Grocholsky, B. (2002). *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney.

[Hoffmann and Tomlin, 2010] Hoffmann, G. M. and Tomlin, C. J. (2010). Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47.

[Howard et al., 2002] Howard, A., Matarić, M. J., and Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems (DARS02)*, Fukuoka, Japan.

[Julian et al., 2011] Julian, B. J., Angermann, M., Schwager, M., and Rus, D. (2011). A scalable information theoretic approach to distributed robot coordination. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. Sumbitted.

[Krause and Guestrin, 2007] Krause, A. and Guestrin, C. (2007). Near-optimal observation selection using submodular functions. In *Proceedings of 22nd Conference on Artificial Intelligence (AAAI)*, Vancouver, Canada.

[Krause et al., 2006] Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. (2006). Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *In Proceedings of Information Processing in Sensor Networks (IPSN)*, Nashville, TN.

[Krause et al., 2008] Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284.

[Li and Cassandras, 2005] Li, W. and Cassandras, C. G. (2005). Distributed cooperative coverage control of sensor networks. In *Proceedings of the IEEE Conference on Decision ans Control, and the European Control Conference*, Seville, Spain.

[Lynch et al., 2008] Lynch, K. M., Schwartz, I. B., Yang, P., and Freeman, R. A. (2008). Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724.

[Martínez, 2010] Martínez, S. (2010). Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Transactions on Control Systems Technology*, 18(2):419–500.

[Palomar and Verdú, 2006] Palomar, D. P. and Verdú, S. (2006). Gradient of mutual information in linear vector gaussian channels. *IEEE Transactions on Information Theory*, 52(1):141–154.

[Palomar and Verdú, 2007] Palomar, D. P. and Verdú, S. (2007). Representation of mutual information via input estimates. *IEEE Transactions on Information Theory*, 53(2):453–470.

[Schwager et al., 2009] Schwager, M., Rus, D., and Slotine, J. J. (2009). Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357–375.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.

[Viola and Wells III, 1997] Viola, P. and Wells III, W. M. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154.

[Vitus and Tomlin, 2010] Vitus, M. P. and Tomlin, C. J. (2010). Sensor placement for improved robotic navigation. In *In the Proceedings of Robotics: Science and Systems*, Zaragoza, Spain.