A Decentralized Control Policy for Adaptive Information Gathering in Hazardous Environments

Philip Dames[1], Mac Schwager[2], Vijay Kumar[1], and Daniela Rus[3]

***Abstract*— This paper proposes an algorithm for driving a group of resource-constrained robots with noisy sensors to localize an unknown number of targets in an environment, while avoiding hazards at unknown positions that cause the robots to fail. The algorithm is based upon the analytic gradient of mutual information of the target locations and measurements and offers two primary improvements over previous algorithms [6], [13]. Firstly, it is decentralized. This follows from an approximation to mutual information based upon the fact that the robots' sensors and environmental hazards have a finite area of influence. Secondly, it allows targets to be localized arbitrarily precisely with limited computational resources. This is done using an adaptive cellular decomposition of the environment, so that only areas that likely contain a target are given finer resolution. The estimation is built upon finite set statistics, which provides a rigorous, probabilistic framework for multi-target tracking. The algorithm is shown to perform favorably compared to existing approximation methods in simulation.**

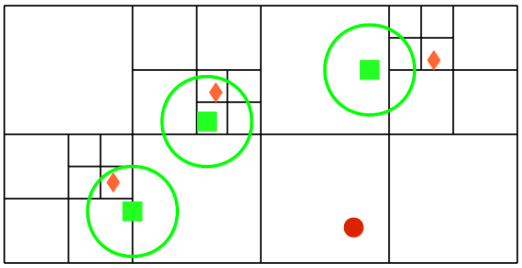


Fig. 1. Illustration of our multi-robot multi-target localization algorithm. The robots (green squares) estimate the locations of targets (orange diamonds) and hazards (red dots) with high resolution by adaptively refining a cellular decomposition of the environment, despite having noisy sensors. The robots move to improve their estimate of the target locations while avoiding the estimated hazard locations by following the gradient of mutual information. The robots' finite sensor footprints (green circles) allow for decentralized estimation and control computations.

I. INTRODUCTION

Teams of mobile robots may be used in many applications to gather information about unknown, hazardous environments, taking measurements at multiple locations while keeping humans out of harm's way. It would be useful, for example, to deploy a team of robots to search for survivors in a building after an earthquake or other disaster, where the number of survivors is unknown a priori. In this scenario the building may be structurally unstable and there may be fires or exposed live electrical wires in the environment, all of which may cause harm to rescuers and robots. As multiple robots will likely fail, it is advantageous to use low-cost platforms. However such platforms have limited capabilities and thus the control strategy should make minimal assumptions about the sensors and environment.

Our approach to tackle this problem employs a coarse, high-level sensor model, wherein sensors only provide binary information indicating whether they have detected a target or not while hazards are only "detected" through robot failures. With such coarse sensing capabilities it is natural to also use a coarse representation of the environment, decomposing the space into a collection of cells. Our algorithm uses a Bayesian estimator over this discrete space to localize the potential hazards and targets (*e.g.,* trapped survivors). Using these estimates the control algorithm moves the team of robots in the direction of greatest immediate information gain, a strategy sometimes called "information surfing" [5]. More precisely, the controller moves the robots along the gradient of mutual information of target locations and measurements with respect to the positions of the robots.

This paper improves upon an algorithm recently proposed by the authors in [13] in two ways. Firstly, the implementation is decentralized (*i.e.,* there is no single point of failure in the system) by introducing a communication protocol where robots share their measurements with one another and by developing an approximation to mutual information. This approximation is based upon the fact that hazards have a finite region of influence in the environment and real sensors are only able to measure objects in a finite subset of the environment, *i.e.,* within the sensor footprint. Secondly, we introduce a method to allow for finer localization of targets with limited computational resources by adaptively tuning the resolution of the environment model. When a region attains a high likelihood of containing a target its resolution is refined. Conversely, a region that reaches a low likelihood of containing a target is given a coarser resolution.

Algorithms that do not take advantage of statistical independences generally have exponential time complexity in the number of robots and the number of cells in the environment, limiting their practical use to a few robots with a relatively coarse environment grid. One focus of this work will be to take advantage of existing independences and to develop principled and numerically validated approximations.

*This work was funded in part by AFOSR Grant FA9550-10-1-0567; ONR Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031; and the SMART Future Mobility project. The first author was supported by an NDSEG fellowship from the Department of Defense.

[1]P. Dames and V. Kumar are with The GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA {pdames,kumar}@seas.upenn.edu

[2]M. Schwager is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA schwager@bu.edu

[3]D. Rus is with The Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, Cambridge, MA 02139, USA, rus@csail.mit.edu.

A. Related Work

There is extensive work on using Bayesian filters to estimate unknown or uncertain environments, much of which Thurn, Burgard, and Fox describe in [15]. Mutual information as a control objective for active estimation has also been widely used. Both Grocholsky in [5] and Bourgault, et al. in [1] use mutual information to drive robots for target tracking and exploration. These works use a Decentralized Data Fusion (DDF) formalism for decentralization and optimize over a discrete set of actions. In [6] Hoffmann and Tomlin use mutual information for control with highly non-Gaussian belief states, achieving scalability by using a pairwise approximation to mutual information. In [7] Julian, et al. use the identical gradient of mutual information as our work to drive a network of robots for general environment state estimation tasks. That work uses a consensus algorithm to achieve decentralization and a sampling strategy to reduce complexity. None of these works consider robot failures due to environmental hazards, as we do here.

In the context of multi-target tracking, we wish to estimate both the number *and* locations of targets within the environment. Many traditional approaches seek to extend single-target filters to this new domain, either by concurrently running many such filters, one for each target believed to exist, or by abstracting the problem to tracking a single meta-target which lives in a higher dimensional state space. However, such approaches run into the issue of data association, *i.e.,* matching measurements to targets, which is computationally expensive for large numbers of targets as there are combinatorially many possible associations. Methods for estimating the data association include maximum likelihood approaches (such as in GraphSLAM) or branch-and-bound (as in the sparse extended information filter) are described in detail by Thrun, Burgard, and Fox in [15]. These have the disadvantage of disallowing the correction of past errors and requiring searches over measurement history, respectively.

Multi-target tracking has also been addressed extensively in the radar tracking community; Pulford provides a taxonomy of techniques in [10]. Out of these we elect to use finite set statistics; see the companion technical report [4] for a quick primer, or the paper [9] or book [8] by Mahler, et al. for more rigorous treatments. This formalism concisely encodes the problem of multi-target tracking without needing to address the problem of data association explicitly. However, the existing work in radar tracking is based on stationary sensors. Ristic, Clark, and Vo in [11], [12] use the FISST framework for tracking multiple targets using Rényi divergence as the objective, rather than mutual information, however this work considers much more capable sensors (*i.e.,* infinite field of view returning both range and bearing). Our work adapts these estimation tools and applies a mutual information based control strategy to drive a group of robots for *active* multi-target tracking and hazard avoidance.

II. PROBLEM FORMULATION

Consider a situation where n robots move in a bounded, planar environment $E \subset \mathbb{R}^2$. The environment is divided into a set of m_T cells, denoted $\{E_j^s\}_{j=1}^{m_T} \subset E$, within which targets may be located. A random finite set (RFS) denoting occupied cells is $X \in \mathcal{X}$, where $\mathcal{X}$ is the set of all possible RFSs for a given discretization. Similarly, another discretization $\{E_j^h\}_{j=1}^{m_H} \subset E$ is used to represent the locations of hazards within the environment and a RFS of cell labels drawn from this discretization is denoted $H \in \mathcal{H}$.

Robot i is at position $q_i^t \in E$ at time t, and the positions of all the robots can be written as the stacked vector $q^t = [(q_1^t)^T \dots (q_n^t)^T]^T$. Each robot is equipped with a binary sensor which gives measurements $z_i \in \{0, 1\}$ indicating whether or not the sensor has detected a target. Robots can also detect the failure status of other robots, $f_i \in \{0, 1\}$, where $f_i = 1$ indicates that robot i has failed. Let the vector of sensor measurements be given by $Z = [z_1, \dots, z_n]^T$, where $Z \in \{0, 1\}^n = \mathcal{Z}$, and the vector of all failure statuses by $F = [f_1, \dots, f_n]^T \in \{0, 1\}^n$.

A. Individual Sensors

As previously mentioned, the robots have a chance of failure due to hazards in the environment. Let the probability of robot i failing due to a hazard in cell E_j^h be modeled by $\mathbb{P}(f_i = 1 \mid j \in H) \approx \alpha(q_i, E_j^h)$ while $\mathbb{P}(f_i = 1 \mid j \notin H) = 0$. We assume that the hazards act independently so

$$\mathbb{P}(f_i = 0 \mid H) = \overline{p}_{\text{f}} \prod_{j \in H} \mathbb{P}(f_i = 0 \mid j), \tag{1}$$

as the only way to not have a failure is to not fail due to any of the individual hazards or due to some other failure with probability $p_{\text{f}} = 1 - \overline{p}_{\text{f}}$ (typically $p_{\text{f}} \ll 1$). The probability of failure is then the additive complement of (1).

When a robot has failed it provides no further information about the location of targets, leading to the conditional probability $\mathbb{P}(z_i = 1 \mid f_i = 1, X) = 0$. If a sensor is still functional, the detection equations are analogous to that of the hazards, beginning with $\mathbb{P}(z_i = 1 \mid f_i = 0, j \in X) \approx \mu(q_i, E_j^s)$ and $\mathbb{P}(z_i = 1 \mid f_i = 0, j \notin X) = 0$. The targets are also assumed to act independently on the sensors so

$$\mathbb{P}(z_i = 0 \mid f_i = 0, X, H) = \overline{p}_{\text{fp}} \prod_{j \in X} \mathbb{P}(z_i = 0 \mid f_i = 0, j), \tag{2}$$

where $p_{\text{fp}} = 1 - \overline{p}_{\text{fp}}$ is the probability of a false positive reading. We defer discussion of the specific forms of the α and μ functions until Sec. II-C.

B. Multiple Sensors

Now we derive expressions for a group of robots together, which will be used in the control law in Sec. IV. Robot failures are assumed to be conditionally independent of one another given the positions of the hazards so that,

$$\mathbb{P}(F \mid X, H) = \prod_i \mathbb{P}(f_i \mid H), \tag{3}$$

where $\mathbb{P}(f_i \mid H)$ comes from (1). Similarly, the robots' sensor measurements are conditionally independent given the locations of the targets, so that

$$\mathbb{P}(Z \mid F, X, H) = \prod_i \mathbb{P}(z_i \mid f_i, X), \tag{4}$$

where $\mathbb{P}(z_i \mid f_i, X)$ comes from (2). Finally, integrating over the possible failure states we get

$$\mathbb{P}(Z \mid X, H) = \prod_i \sum_{f_i \in \{0,1\}} \mathbb{P}(z_i \mid f_i, X)\mathbb{P}(f_i \mid H). \quad (5)$$

C. Sensor and Failure Likelihoods

Here we develop the failure model, $\mathbb{P}(f_i = 1 \mid j \in H)$, and sensor model, $\mathbb{P}(z_i = 1 \mid f_i = 0, j \in X)$, of the robots in more detail. Firstly, to take into account the finite footprints these models should have compact support. Let F_i be the set of labels of cells within the footprint of robot i and consider the subset of RFSs containing targets in F_i, $\mathcal{V}_i = \{X \in \mathcal{X} \mid x \in F_i \ \forall x \in X\}$. This is found using the projection $r_T : \mathcal{X} \to \mathcal{V}_i$ given by $r_T(X) = X \cap F_i$. Note that this map is surjective but not injective as long as F_i is a proper subset of E, so no inverse mapping exists. However we may still define the right inverse, where $r_T(r_T^{-1}(V_i)) = V_i$ but $r_T^{-1}(r_T(X)) \neq X$. The right inverse of the projection is $r_T^{-1}(V_i) = \{X \mid r_T(X) = V_i\}$, which returns multiple values in general. Let W_i be the analogous neighborhood in the hazard grid with projection r_H.

Secondly, the features may be located anywhere within the cell. Given this, the probability of failure due to a hazard in cell E_j^h is given by

$$\alpha(q_i, E_j^h) = \int_{E_j^h} g_h(q_i, x)\mathbb{P}(x)\, dx \approx \frac{1}{m_j}\sum_{k=1}^{m_j} g_h(q_i, e_{j,k}^h), \quad (6)$$

where $g_h(q_i, x)$ is a function describing the probability of failure due to a hazard at location x and $\mathbb{P}(x)$ is a distribution of the location of the hazard in the cell. We approximate this integral by a sum over a set of m_j points in the cell, $\{e_{j,k}^h\}_{k=1}^{m_j} \in E_j^h$, and given no available information beyond our binary failure readings, we let the distribution over these points be uniform. While the simplest approach is to use the cell centroids, we allow for multiple points when the footprint is small compared to the size of a cell.

Analogously the probability of detection is

$$\mu(q_i, E_j^s) \approx \frac{1}{m_j}\sum_{k=1}^{m_j} g_s(q_i, e_{j,k}^s), \quad (7)$$

where we again approximate the expectation by a sum over a set of points $\{e_{j,k}^s\}_{k=1}^{m_j} \in E_j^s$. This integration over the cell naturally takes into consideration the amount of the cell that is visible to the sensor: if only a small portion is visible then μ will be low as most terms in the sum will be zero, while if the robot can see most of the cell then μ will be larger. However, it does not take into account the area viewed during previous time steps, an issue for exploration over discrete spaces which may lead to falsely determining that a cell is empty, as noted by Waharte, et al in [16].

D. Communication

A communication protocol is necessary in order to decentralize the exploration task. We take a simple approach to the problem, using the standard disk model as outlined in Alg. 1. Note that it is not necessary to send the failure status of previous times, as the robot would be unable to communicate if it had failed prior to the current time t. A justification for the incorporation of old measurements is given in Sec. III.

Algorithm 1 Communication

```
for All robots, i do
    Discover robots in communication range, N_i
    for j ∈ N_i do
        Lookup time of last communication, τ_j
        if τ_j < t then
            Send q_i^{τ_j:t}, z_i^{τ_j:t}, f_i^t
            Receive q_j^{τ_j:t}, z_j^{τ_j:t}, f_j^t
            τ_j ← t
        end if
    end for
    Update Bayesian filter using new measurements
end for
```

III. BAYESIAN ESTIMATION

As the sensors explore the environment and exchange measurements, a recursive Bayesian filter makes use of the collected information in order to estimate the target and hazard locations. Let $\varphi^t(X) = \mathbb{P}(X \mid Z^{1:t}, F^{1:t})$ be the estimated distribution over RFSs of target positions at time t and $\psi^t(H) = \mathbb{P}(H \mid Z^{1:t}, F^{1:t})$ be the estimate of the hazard RFS distribution. In [13] Theorem 1 we showed that the posterior estimates for the targets and the hazards are probabilistically independent, so that $\mathbb{P}(X, H \mid Z^{1:t}, F^{1:t}) = \varphi^t(X)\psi^t(H)$. A separate filter can therefore be updated for each, according to the Bayesian update equations

$$\varphi^t(X) = \frac{\mathbb{P}(Z^t \mid F^t, X)\varphi^{t-1}(X)}{\sum_{\mathcal{X}} \mathbb{P}(Z^t \mid F^t, X)\varphi^{t-1}(X)}, \quad (8)$$

$$\psi^t(H) = \frac{\mathbb{P}(F^t \mid H)\psi^{t-1}(H)}{\sum_{\mathcal{H}} \mathbb{P}(F^t \mid H)\psi^{t-1}(H)}. \quad (9)$$

In this section we decentralize these filters by separating them into updates over individual measurements. This way each robot maintains a separate filter and is able to incorporate past measurements. Furthermore, this iterative update reduces the complexity of the Bayesian update to be linear, rather than exponential, in the number of measurements.

Note that we are using the Markov assumption in order to perform filtering, *i.e.,* that the past and future are conditionally independent given the present. Furthermore, since the environment is static (*i.e.,* the belief will not change if the robots cease taking measurements) and we have assumed conditional independence of robots given the target and hazard locations, the current belief can be written as

$$\phi^t(X) \propto \prod_i \prod_{t=1}^{\tau_i} \mathbb{P}(z_i^t \mid f_i^t, X)\phi^0(X). \quad (10)$$

Thus the filtering approach described in Alg. 1 will result in the same posterior regardless of the order in which individual updates are applied.

Now that we have reduced the Bayesian filter problem to an iterative update over individual measurements, the form of these individual updates must be discussed. Each of these updates can leverage the fact that each robot only sees a subset of the environment, as described in Sec. II-C.

Theorem 1: The Bayesian update over the full environment can be computed from the Bayesian update over the neighborhood $\mathcal{V}_i$ as

$$\varphi^t(X) = \frac{\varphi^t(V_i)}{\varphi^{t-1}(V_i)}\varphi^{t-1}(X). \tag{11}$$

Proof: See companion technical report [4]. ■

The hazard updates can be similarly decomposed using the projection $r_H : \mathcal{H} \to \mathcal{W}_i$. Statistics of interest of these distributions include the probability of n targets in the environment, $\mathbb{P}(|X| = n) = \sum_{X||X|=n} \varphi(X)$, and, as a special case of (11), the probability of an individual cell i being occupied,

$$\varphi(i \in X) = \sum_{X|i\in X} \varphi(X). \tag{12}$$

IV. MUTUAL INFORMATION GRADIENT CONTROLLER

In this section we derive an information seeking controller using the analytic gradient of mutual information previously given in [13]. The mutual information of two random variables is an information theoretic quantity ([3], [14]) that describes the amount of information that can be gained about one random variable (*e.g.,* targets) by observing another (*e.g.,* measurements). Mutual information is defined as

$$I[\mathcal{X}, \mathcal{Z}] = \int_{\mathcal{X}}\int_{\mathcal{Z}} \mathbb{P}(X, Z) \log \frac{\mathbb{P}(X, Z)}{\mathbb{P}(X)\mathbb{P}(Z)}\, dZ\, \delta X. \tag{13}$$

Here the information is written as though $\mathcal{X}, \mathcal{Z}$ were continuous random variables, however equivalent expressions can be written for the discrete case. The integral has also been replaced by a set integral as we have a distribution over random finite sets. The key to deriving our controller is to note that this information depends upon the locations of the robots, a deterministic quantity. To indicate this dependence we use a q following a semicolon, writing the mutual information as $I[\mathcal{X}, \mathcal{Z}; q]$. Then the gradient of mutual information with respect to the parameter q is given in the following Theorem.

Theorem 2: Let random vector $\mathcal{Z}$ and random finite set $\mathcal{X}$ be jointly distributed with distribution $\mathbb{P}(\mathcal{X}, \mathcal{Z}; q)$ that is differentiable with respect to the parameter vector $q \in \mathbb{R}^{2n}$ over $E^n \subset \mathbb{R}^{2n}$. Also, suppose that the support $\mathcal{X} \times \mathcal{Z}$ of $\mathbb{P}(\mathcal{X}, \mathcal{Z}; q)$ does not depend on q. Then the gradient of mutual information with respect to the parameters q over E^n is

$$\frac{\partial I[\mathcal{X}, \mathcal{Z}; q]}{\partial q} = \iint_{\mathcal{Z},\mathcal{X}} \frac{\partial \mathbb{P}(X, Z; q)}{\partial q} \log \frac{\mathbb{P}(X, Z; q)}{\mathbb{P}(X)\mathbb{P}(Z; q)}\, \delta X dZ. \tag{14}$$

Proof: See Theorem 2 in [13]. ■

A. Finite Footprint Approximation

We leverage the fact that sensors have a finite footprint in order to derive an approximate decoupling of the mutual information and reduce the complexity of the gradient calculations. We begin by defining an undirected graph where each sensor is a node and an edge exists between nodes i, j if their sensor footprints overlap, ie. $F_i \cap F_j \neq \varnothing$. For example, the configuration in Fig. 2 represents a graph with edges (1,4) and (2,4). A *coalition* of sensors is a connected component of this graph, denoted C_i, which can be computed using Alg. 2. The use of a subscript C_i represents the union of that quantity over the coalition C_i, for example the set of all tuples of measurements is $\mathcal{Z}_{C_i}$ and the footprint is F_{C_i}.

Algorithm 2 Coalition Identification For Robot i

$Q_i^k = \{q_j^t \mid \|q_j^t - q_i^t\| \leq R_c, F_i \cap F_j \neq \varnothing\}$
$k = 0$
repeat
  $k \leftarrow k + 1$
  $Q_i^k \leftarrow Q_i^{k-1}$
  for $j \mid \|q_j^t - q_i^t\| \leq R_c$ **do**
    $Q_i^k \leftarrow Q_i^k \cup Q_j^k$
  end for
until $Q_i^k = Q_i^{k-1}$

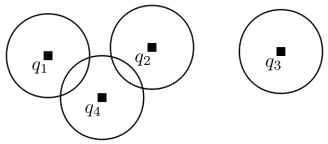


Fig. 2. In this situation the team of robots, with their footprints shown by the circles, is divided into two coalitions $C_1 = \{1, 2, 4\}, C_2 = \{3\}$.

Using the the chain rule for mutual information given by Cover in [3],

$$I[\mathcal{X}, \mathcal{Z}] = \sum_i I[\mathcal{X}, \mathcal{Z}_{C_i} \mid \mathcal{Z}_{C_1}, \ldots, \mathcal{Z}_{C_{i-1}}]. \tag{15}$$

We then make the assumption that observations of separate coalitions are independent of one another (*not* conditionally independent, as before) so that mutual information is

$$I[\mathcal{X}, \mathcal{Z}] \approx \sum_i I[\mathcal{X}, \mathcal{Z}_{C_i}] = \sum_i I[\mathcal{V}_{C_i}, \mathcal{Z}_{C_i}]. \tag{16}$$

Note that the equality is exact, as measurements by a coalition of sensors are independent of targets outside the coalition's footprint. The error in our approximation is

$$I[\mathcal{X}, \mathcal{Z}] - \sum_i I[\mathcal{V}_{C_i}, \mathcal{Z}_{C_i}] = H[\mathcal{Z}] - \sum_i H[\mathcal{Z}_{C_i}], \tag{17}$$

which is bounded from above by zero with equality iff the measurements are independent, as shown by Cover in [3].

Similarly, the gradient in (14) can be written as

$$\frac{\partial I[\mathcal{X}, \mathcal{Z}; q]}{\partial q} \approx \sum_i \frac{\partial I[\mathcal{V}_{C_i}, \mathcal{Z}_{C_i}; q]}{\partial q}. \tag{18}$$

Note that the gradient of mutual information with respect to the position of each sensor depends only upon the locations of the sensors in its coalition.

The intuition behind this is that robots near each other should coordinate their motions to better localize targets while robots that are sufficiently far apart can act as independent agents with little penalty to performance. Using this reasoning it is simple to adapt the definition of a coalition to fit the computational budget of a particular robot: coordinate with robots that are increasingly separated until a maximum number is reached or no other robots are within communication range. This approach is similar in spirit to the single- and pairwise-node approximations presented by Hoffmann and Tomlin in [6], where mutual information is approximated by the sum of mutual information from each sensor or each pair of sensors in the network. However our approach offers a systematic approach of how to best spend the computational resources available to each robot.

B. Control Law

Writing the gradient from (14) in terms of known quantities, we have

$$\frac{\partial I[\mathcal{V}_C, \mathcal{Z}_C; q]}{\partial q} = \sum_{Z_C \in \mathcal{Z}_C} \sum_{V \in \mathcal{V}_C} \sum_{W \in \mathcal{W}_C} \frac{\partial \mathbb{P}(Z_C \mid V, W; q)}{\partial q} \varphi^t(V) \psi^t(W) \times \log \frac{\sum_{W \in \mathcal{W}_C} \mathbb{P}(Z_C \mid V, W; q) \psi^t(W)}{\sum_{V \in \mathcal{V}_C} \sum_{W \in \mathcal{W}_C} \mathbb{P}(Z_C \mid V, W; q) \varphi^t(V) \psi^t(W)}. \tag{19}$$

Here $\varphi^t(V)$ and $\psi^t(W)$ comes from (8) and (9) and $\mathbb{P}(Z_C \mid V, W; q)$ from (5). The gradient of (2) for a single robot is

$$\begin{aligned} \frac{\partial \mathbb{P}(z_i = 0 \mid V, W; q)}{\partial q} &= -\mathbb{P}(f_i = 0 \mid W)\mathbb{P}(z_i = 0 \mid V, W) \\ &\quad \times \sum_{j \in V} \frac{1}{1 - \mu(q_i, E_j^s)} \frac{\partial \mu(q_i, E_j^s)}{\partial q_i} \\ &\quad + \mathbb{P}(f_i = 0 \mid W)\mathbb{P}(z_i = 1 \mid V, W) \\ &\quad \times \sum_{k \in W} \frac{1}{1 - \alpha(q_i, E_k^h)} \frac{\partial \alpha(q_i, E_k^h)}{\partial q_i}, \end{aligned} \tag{20}$$

and when $z_i = 1$ it is simply the negative of this. The derivative of (5) is found using (20) and the chain rule.

Using these results, our proposed controller is given by

$$q_i^{t+1} = q_i^t + k \frac{\frac{\partial I[\mathcal{V}_C, \mathcal{Z}_C; q^t]}{\partial q_i}}{\left\| \frac{\partial I[\mathcal{V}_C, \mathcal{Z}_C; q^t]}{\partial q_i} \right\| + \epsilon}, \tag{21}$$

where $i \in C$, k is the maximum step size, and $\epsilon \ll 1$ avoids singularities near critical points. It is important to note that this is not a traditional gradient ascent controller, as mutual information changes as measurements and failures are incorporated into the target and hazard beliefs. Also, hazard avoidance is implicitly built into the proposed controller as the information gained by a failed sensor is zero so that robots naturally avoid areas where they expect to fail.

V. Adaptive Cellular Decomposition

As can be seen from (19), the number of computations involved in the mutual information gradient for a single robot in coalition C is $O(2^{|C|}|\mathcal{V}_C||\mathcal{W}_C|)$. This is exponential in the number of robots in a coalition and linear in the number of possible RFSs in the coalition footprint. Both depend on the size of the footprint and maximum number of targets as

$$|\mathcal{V}_C| = \sum_{k=0}^{|X|_{\max}} \binom{|F_C|}{k}. \tag{22}$$

So $|\mathcal{V}_C|$ is $O(|F_C|^{|X|_{\max}})$ when $|X|_{\max} \ll |F_C|$ and $O(2^{|F_C|})$ when $|X|_{\max} \approx |F_C|$. Note that while limiting the possible number of targets, $|X|_{\max}$, decreases the computational complexity, it adds dependence between the measurements of the agents. This is because a detection in one region of the environment means that a target is more likely in that region *and* less likely to be in another region. If this were not bounded then the inequality in (16) would become an equality.

While the exact complexity depends upon the current configuration of the team of robots as well as the representation of the environment, we can examine two limiting cases. The complexity is lower bounded when each robot is in its own coalition, which is $O(|\mathcal{V}_i||\mathcal{W}_i|)$. On the other extreme, the complexity is upper bounded when all robots are in a single coalition, in which case it is $O(2^n|\mathcal{V}_C||\mathcal{W}_C|)$. Despite the lack of guarantee of reduced complexity, empirically we have seen improved performance using the sensor grouping. Also, the benefits will tend to increase as the size of the environment increases because it is also more likely for robots to split into separate coalitions.

A. Adaptive Cellular Algorithm

In order to store the full distribution over RFSs for a large-scale environment, the total number of cells used must be kept at a tractable level. We do this using an adaptive cellular decomposition based on the quadtree data structure, however the methodology can be easily extended to work with other decompositions. Quadtrees have been used in other localization tasks, such as the work of Carpin, et al. in [2], with the difference being that our implementation allows for the removal of leaves from the tree.

The main idea is that initially a coarse discretization is used and refined only in areas that are likely to contain a feature. If the detection turns out to be a false positve, the procedure can then be reversed. The two basic operations of this adaptive cellular decomposition are the addition and removal of a cell, given in Alg. 3 and Alg. 4, respectively. In Alg. 4, $r_j(X) : \mathcal{X} \to \mathcal{V}_{j^\complement}$ is a projection onto the complement of cell j, $j^\complement$.

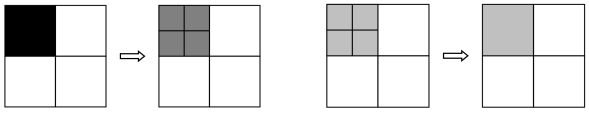
These two operations can then be used to adapt any cellular decomposition of the environment. A refinement procedure involves removing cells that are occupied with sufficiently high probability, $\varphi(i \in X) > \tau_o$ for some threshold τ_o, and then adding some number of child cells, four in the case of a quadtree. This is illustrated in Fig. 3a.

Algorithm 3 Add Cell

$\mathcal{X}' \leftarrow \mathcal{X}$
for $X \in \mathcal{X} \mid |X| <$ max number of targets **do**
  $\varphi'(X) \leftarrow \frac{1}{2}\varphi(X)$
  $\varphi'(X \cup \{m_T + 1\}) \leftarrow \frac{1}{2}\varphi(X)$
  $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{X \cup \{m_T + 1\}\}$
end for
$m_T' \leftarrow m_T + 1$

Algorithm 4 Remove Cell

for $V \in \mathcal{V}_{j\mathfrak{C}}$ **do**
  $\varphi(V) \leftarrow \sum_{X|r_j(X)=V} \varphi(X)$
  $\mathcal{X}' \leftarrow \mathcal{X}' \setminus \{V \cup \{j\}\}$
end for
$m_T \leftarrow m_T - 1$



(a) Cell refinement (b) Cell merge

Fig. 3. A simple 2×2 grid example where the shading indicates the probability that a cell is occupied with white being 0 and black being 1. A cell refinement procedure is shown in (a), where a large occupied cell is divided into four smaller cells with uniform occupancy probability. A grid merging procedure is shown in (b), where four empty sub-cells with the same parent cell are merged to form the parent cell.

Similarly the cell merging procedure, illustrated in Fig. 3b, takes place if all children of a single parent are occupied with sufficiently low probability, *i.e.,* $\varphi(j \in X) < \tau_e < \tau_o$ for all children j of a single parent cell. All children are removed and then the parent is added. Note that the distribution used in the Bayesian filter is over RFSs, which is then used to calculate the occupancy probability of individual cells used in the grid adaptation via (12).

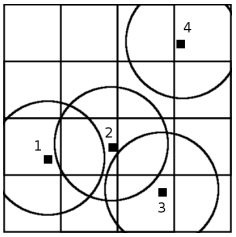
VI. SIMULATION AND RESULTS

Looking at the error due to the finite footprint approximation given in (17), it is difficult to gain intuition as to how tight of a bound this is and how much the approximation affects the control decision. Also, as discussed in Sec. V, the computational complexity is difficult to quantify as it depends upon the relative configuration of the robot team. To better understand these quantities we run a series of simulations to illustrate the performance of several approximations:

- Single-node (SN): robots act independently $O(|\mathcal{V}_i||\mathcal{W}_i|)$
- Pairwise (PW): robots consider pairwise information with each other robot $O(\sum_{j \neq i} |\mathcal{V}_i \cup \mathcal{V}_j||\mathcal{W}_i \cup \mathcal{W}_j|)$
- Footprint intersection (FI): robots consider only other robots whose footprints intersect their own (*i.e.,* redefine coalitions from Sec. IV-A to be neighbors in the graph)
- Finite-footprint (FF): described in Sec. IV-A

The SN and PW approximations were proposed by Hoffmann and Tomlin in [6].

For the comparative test, we initialize a robot team in the configuration specified in Fig. 4 and consider the control direction computed by robot 1. This configuration is chosen so that the robot under consideration belong to a different coalition (or set of coalitions) for each approximation method, as listed in Fig. 4. Table I contains the mean computational time for each method. As is expected the full computation takes considerably longer than any of the approximations, with our approximations performing favorably with respect to existing approximation techniques.



Method	Coalitions For Robot 1
Full	$\{1,2,3,4\}$
SN	$\{1\}$
PW	$\{1\},\{1,2\},\{1,3\},\{1,4\}$
FI	$\{1,2\}$
FF	$\{1,2,3\}$

Fig. 4. The locations of the robots in the test environment, given by the black squares with their sensor footprints indicated by the circles, are shown on the left. The coalitions containing robot 1 are shown in the table on the right for each approximation method.

TABLE I
COMPARISON OF APPROXIMATION METHODS FOR UNIFORM BELIEF

Method	Full	SN	PW	FI	FF
Time (s)	0.2061	0.0201	0.0705	0.0217	0.0346
Mean Error (°)	N/A	-10.848	-0.500	-0.501	0.492

As can be seen from (19), the direction of the gradient depends upon the current belief about the environment. To gain intuition about the performance of these approximations we consider two cases. The first case is a near uniform distribution, with small random perturbations. We performed 500 trials, calculating the direction of the gradient vector and the error relative to the full mutual information computation. The mean errors, given in Table I, are relatively small, except when using SN, and fairly consistent with standard deviations on the order of $0.1°$ for each approximation.

To further sample the space of distributions, for each possible RFS containing at most five targets we initialize a distribution that has nearly converged to that RFS by setting 95% of the probability mass to that distribution and the remainder uniform across all other RFSs. Box plots of the error in the computed control direction for each approximation method are shown in Fig. 5. As the figure shows, SN performs poorly with a large spread in the direction error and relatively little probability mass near zero error. The PW and FI approximations perform comparably, with opposite biases in the distribution. PW is more peaked near zero error, but also has more outliers than FI. FF performs the best out of all four methods, with the most peaked distribution near zero error and fewest outliers. The maximum errors are $136.1°, -115.2°, -84.95°, -65.3°$ for SN, PW, FI, and FF respectively. Such large errors occur when the maximum likelihood target locations fall within the footprints of other robots in the group, in which case the decision made by robot 1 will be more strongly affected by the motion of the other robots in the team.

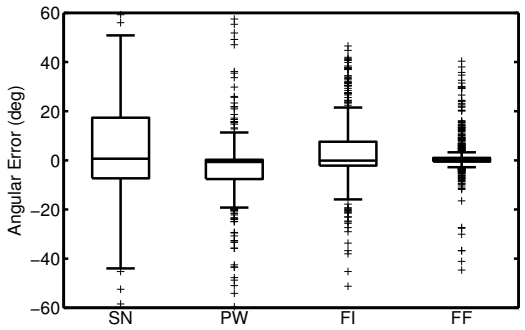


Fig. 5. Box plots showing the error in angle of the gradient approximations for each of the approximation methods, measured in degrees. Each box plot represents $\sim$ 7000 data points. Not shown are 15.8%, 1.80%, 0.08%, and 0.04% of the data, for the SN, PW, FI, and FF respectively, which correspond to large control deviations.

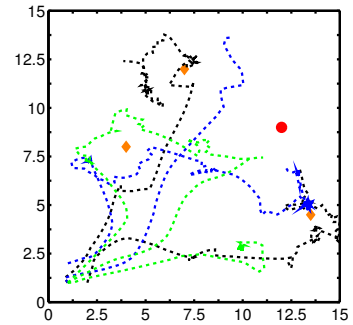
In order to test the performance of our proposed algorithm in terms of localizing targets, we conduct a series of simulations over the environment shown in Fig. 6a, along with typical paths taken by the robots. The sensor model used is

$$g_s(q_i, x) = \begin{cases} \overline{p}_{\text{fn}} \exp\left(\frac{-\|q_i - x\|^2}{\sigma^2}\right), & \|q_i - x\| \leq R \\ 0, & \|q_i - x\| > R \end{cases}$$

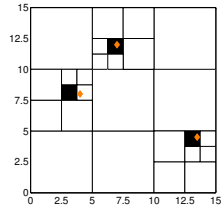
where $p_{\text{fn}} = 0.05, \sigma = 2$, and $R = 6$. The failure model g_h is of the same form, with $p_{\text{s}} = 0.1, \sigma = 1.5$, and $R = 2$. The final estimates of the target and hazard occupancy grids for the given path are shown in Fig. 6b and Fig. 6d, respectively, with a maximum of four targets and two hazards allowed in the environment. If a robot fails, a new one is sent out from a base station located at $(1, 1)$ in the environment. The entropy in the target location estimate, shown in Fig. 6c, generally decreases, with increases due to the addition of cells as well as false positive and missed detections.

VII. CONCLUSIONS

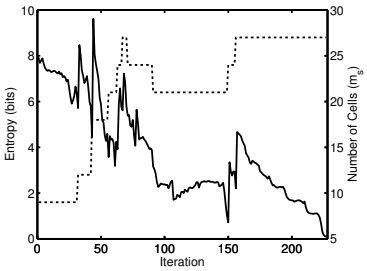
In this paper we proposed an approximation of our multi-robot control policy from [13] based on finite set statistics that allows for decentralization and significant reductions in the computational complexity. This is based on the fact that real sensors and hazards have a limited range of influence in the environment, thus mobile sensors need only consider their local beliefs about the environment and the actions of nearby robots when planning. This intuition is validated through simulations showing that the control error in our approximation is small in most cases. Despite minimal data from noisy, binary sensors and failures, the team of robots is able to localize an unknown number of targets while avoiding hazards. Recursive Bayesian filters maintain the robots' beliefs about the environment while the robots follow the gradient of mutual information, locally maximizing the expected information gain at each time step.



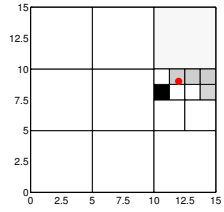
(a) Environment and robot paths



(b) Final target estimate



(c) Target entropy and number of cells



(d) Final hazard estimate

Fig. 6. (a) Sample results in the trial environment. Target locations are given by the orange diamonds and hazard locations by the red square. (b,d) The shading in the cells represents $\varphi(j \in S), \psi(j \in H)$, where white is 0 and black is 1. (c) The solid line is the entropy of $\varphi(S)$ and the dashed line is the number of cells in the target discretization, m_s.

REFERENCES

[1] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE Int. Conf. Intell. Robots and Syst. (IROS)*, pages 540–5, 2002.

[2] S. Carpin, D. Burch, and T.H. Chung. Searching for multiple targets using probabilistic quadtrees. In *IEEE Int. Conf. Intell. Robots and Syst. (IROS)*, pages 4536–43, 2011.

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

[4] P. Dames, M. Schwager, V. Kumar, and D. Rus. A decentralized control policy for adaptive information gathering in hazardous environments. Technical report, University of Pennsylvania, 2012. Available at http://seas.upenn.edu/~pdames.

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

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

[7] B. J. Julian, M. Angermann, M. Schwager, and D. Rus. A scalable information theoretic approach to distributed robot coordination. In *IEEE Conf. Intell. Robots and Syst. (IROS)*, 2011.

[8] R. Mahler. Statistical multisource-multitarget information fusion. 2007. *Artech House, Norwood, MA*.

[9] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.*, 39(4):1152–78, 2003.

[10] GW Pulford. Taxonomy of multiple target tracking methods. *IEE Proc. Radar, Sonar and Navigation*, 152(5):291–304, 2005.

[11] B. Ristic and B.N. Vo. Sensor control for multi-object state-space estimation using random finite sets. *Automatica*, 46(11):1812–8, 2010.

[12] B. Ristic, B.N. Vo, and D. Clark. A note on the reward function for phd filters with sensor control. *IEEE Trans. Aerosp. Electron. Syst.*, 47(2):1521–9, 2011.

[13] M. Schwager, P. Dames, D. Rus, and V. Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Int. Symp. Robotics Research (ISRR)*, August 2011.

[14] C. E. Shannon. A mathematical theory of communication. *Bell Systems Tech. J.*, 27:379–423, 1948.

[15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[16] S. Waharte, A. Symington, and N. Trigoni. Probabilistic search with agile uavs. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2840–5, 2010.