Path Planning for Information Gathering with Lethal Hazards and No Communication

Michael Otte[1(✉)] and Donald Sofge[2]

[1] University of Maryland, College Park, MD 20742, USA
otte@umd.edu
[2] U.S. Naval Research Laboratory, Washington, DC 20375, USA

Abstract. We consider a scenario where agents search for targets in a hazardous environment that prevents communication. Agents in the field cannot communicate, and hazards are only directly observable by the agents that are destroyed by them. Thus, beliefs about hazard locations must be inferred by sending agents to travel along various paths and then observing which agents survive. In other words, agent survival along a path can be used as a sensor for hazard detection; we call this form of sensor a "path-based sensor". We present a recursive Bayesian update for path-based sensors, and leverage it to calculate the expected information gained about both hazards and targets along a particular path. We formalize the resulting iterative information based path planning problem that results from this scenario, and present an algorithm to solve it. Agents iteratively foray into the field. The next path each agent follows is calculated to maximize a weighted combination of the expected information gained about targets and hazards (where the weighting is defined by user preferences). The method is evaluated in Monte Carlo simulations, and we observe that it outperforms other techniques.

1 Introduction

We are motivated by the following scenario: Autonomous agents are used to help search for human survivors ("targets") in a hazardous environment, *but wireless communication is prohibited*. As agents gather information about survivors' whereabouts, they must physically visit special "uplink sites" to upload their information for use by humans and other agents (uplink sites may be, e.g., naval ships or bases). Information gathered by a particular agent is *lost* if that agent is destroyed before reaching an uplink site. Hazards exist in the environment but are invisible. An agent cannot upload data about direct *positive* hazard observations because the only way to positively "observe" a hazard is to be destroyed by it. Luckily, *indirect* information about hazards can be inferred by remembering which path an agent plans to take, and then observing whether or not the agent survives a journey along that path. Agents are less likely to return from paths containing adversaries than paths that are adversary free. Thus, we

M. Morales et al. (Eds.): WAFR 2018, SPAR 14, pp. 389–405, 2020.
https://doi.org/10.1007/978-3-030-44051-0_23

can use path traversal as a sensor for detecting hazards, albeit indirectly—we call this form of sensor a "path-based sensor".

In this paper we show how observations from path-based sensors can be used in a recursive Bayesian framework to refine our beliefs about whether or not hazards exist at various locations in the environment. We also show how to calculate the expected information gain that will result from sending an agent along a *particular* path and then observing whether or not it survives. *Expected information gain* is defined as the expected reduction in the Shannon entropy of our beliefs integrated over the distribution of possible events. In other words, we seek paths that maximize *mutual information* [2,14] between sensor readings and our existing beliefs regarding hazards and targets (Fig. 1).

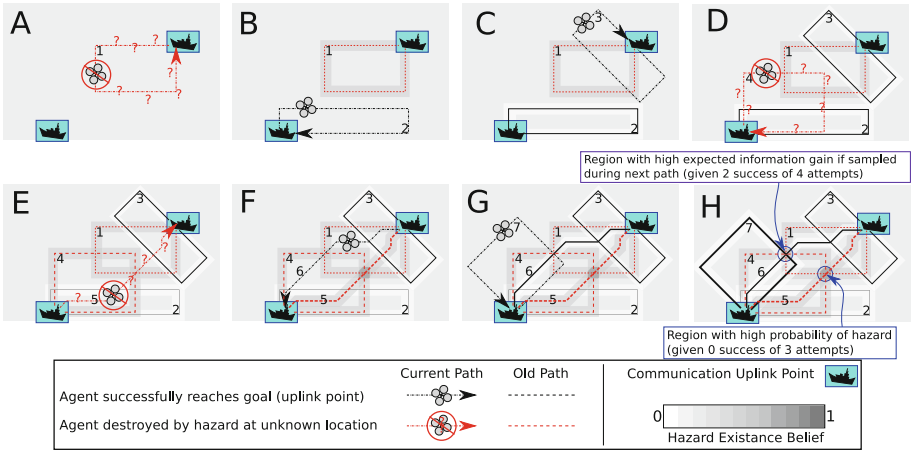

Fig. 1. A-G: search agents attempt seven paths ending at two uplink points (communication is impossible elsewhere). Three are destroyed at unknown locations (A, D, E). Each agent's path, combined with the observation of whether or not that agent survived, is a hazard sensor. H: the maximum information is gained by visiting a location with 50% hazard probability (and not the location with the highest hazard probability). On-board sensors simultaneously detect targets (e.g., human survivors, not shown).

The path-based sensor Bayesian update and expected information gain can be used to plan paths that maximize: (A) the expected information gained about hazard locations; (B) the expected information gained about *target* locations while accounting for the agent's survival given our belief about hazards; and (C) a weighted linear combination of (A) and (B) as specified by a user.

The path-based sensor may produce false positives and/or false negatives. False positives are possible if the agent malfunctions for reasons unrelated to hazards. False negatives are possible because hazards may not destroy every agent they encounter. We assume that agents malfunction every time they move with a known probability $p_{\text{malfunc}} \in [0, 1)$, and that hazards destroy agents they

encounter with a known probability $p_{\text{kill}} \in (0, 1]$. If a hazard fails to destroy an agent, then the hazard remains unobserved by that agent.

The recursive path-based sensor update and information theoretic equations that we derive in this paper can be used to solve a variety of problems. A path-sensor update is the appropriate tool to use whenever a vehicle follows a path carrying a "one-time-use" sensor and yet when we have no direct way of observing when the sensor is triggered. Other scenarios of this form include: radiation or chemical source localization using single-use dosimeters that can be replaced at "safe locations" (e.g., dosimeters that change color when encountering a particular chemical as a result of a chemical reaction); presence mapping of ecological species, geological minerals, etc. where multiple specimens are collected and mixed in a single container that must be returned before its contents can be analyzed; a variety of other search and rescue settings (for example, in a civilian setting targets, hazards, and uplink points may be the survivors of a nuclear disaster, radiation sources, and a fixed communication links, respectively).

The rest of this paper is organized as follows: Related Work appears in Sect. 2. Nomenclature is introduced in Sect. 3. The recursive Bayesian update for the path-based sensor and the information theoretic calculations are described in Sect. 4. The problem we solve is formally defined in Sect. 5, and an algorithm to solve it is described in Sect. 6. In Sect. 7 we run a number of Monte Carlo simulations to evaluate our algorithm and compare it to other methods that have previously been used to solve related problems. Our conclusions are presented in Sect. 8.

2 Related Work

Surveys of previous work on target search can be found in [1] and [11]. The main difference between our work and previous work is the scenario considered: path planning to maximize information gain about targets and hazards in an environment where hazards can destroy agents and communication is impossible. Most previous work investigating target search in hazardous environments has assumed agents are able to communicate from any location in the environment.

The approach presented in [13] shares two important similarities with our work: recursive Bayesian filters are used to update estimates of both target and hazard locations, and the probability that an agent may be destroyed at different locations is used for calculating the mutual information that can be gained about *targets*. Unlike the problem we consider, [13] assumes that robots can always communicate, which makes agent failure locations directly observable. Our assumption that agents cannot communicate from the field means it is impossible to know where agents are destroyed; and therefore we must use a path-based sensor to gather indirect evidence of hazard positions.

Other differences between our work and [13] include: (i) we are interested in planning *multi-step* paths subject to fuel constraints (while [13] uses *information*

surfing[1], a greedy 1-step look-ahead approach that does not consider fuel constraints); (ii) we explore a family of techniques that enable path optimization for any weighted combination of the expected information gained about targets and hazards[2]; (iii) a continuous space formulation is used in [13], and targets are assumed to emit a random signal that is a function over the entire search space. We consider a discrete formulation in which each map cell either does or does not contain a target and/or hazard.

The target information gathering control problem is studied in [9]; mutual information associated with (only) an environmental monitoring mission is considered and information is gathered about environmental state without the possibility of agent failures. The first formal derivation of the gradient of mutual information is also presented in [9], and it is proven that a multi-agent control policy of gradient ascent will converge (gather all information), in the limit. The authors consider a multi-agent mutual information target-localization control [3], again using 1-step look-ahead information surfing. In [5] the method from [3] was implemented on a test-bed and used to demonstrate localization of magnetic sources by quad-rotor aircraft. Two of the authors present a multi-agent information gathering control policies in [4], using both 1-step look-ahead and a 3-step receding horizon control. The main focus of [3–5] is a decentralized multi-agent control formulation. Our work differs from the aforementioned works [3–5,9] in that we consider a scenario in which hazards can destroy agents.

Receding horizon control for gathering information about a moving target following a random walk in the underwater domain is presented in [8]. Similarities to our work include an assumption that long breaks in communication may occur (in [8] this happens when agents are submerged). Our work differs from [8] both in the scenario considered ([8] considers a moving target and assumes false positives are negligible), and in the type of solution that is computed ([8] focuses on distributed multi-agent control—testing horizons of 1 and 4).

A number of other works consider the risk of agent loss within a search or target tracking task. A branch-and-bound technique is used to find paths for search over graphs that attempt to simultaneously optimize vs. cost functions defined over fuel constraints, time, risk, and (optionally) the path endpoint [12]. A neural network based strategy is proven to be robust to partial loss of UAVs; individual planners maximize a heuristic function while learning how other agents tend to behave in various situations [17]. In [7] threats are considered and the robot moves in order to decrease a custom heuristic value based on a probability map, while preliminary work by the same authors that does not consider threats appears in [6]. In [16] agents consider other vehicles to be soft obstacles that present dynamic threats, and use "approximate" dynamic programming to plan movements that maximize a heuristic combination of target confirmation,

[1] In "information surfing" methods an/the agent(s) continually moves "up" a gradient of mutual information using a greedy 1-step look-ahead.

[2] Although [13] uses hazard probabilities to help calculate the expected information gain regarding targets, hazard information is not directly considered as part of the objective function.

environment exploration, and threat avoidance given updated values of threat probability, target probability, and the certainty of these beliefs. Agent loss in hazardous environments with moving (and hostile) targets is considered in [15]. Targets are assumed to move at a constant speed, and agents use a sweeping formation that is adjusted as agents are destroyed. In [10] the probability that at least k of n robots survive a hazardous multipath is maximized.

3 Nomenclature

The search space is denoted $\mathbf{S}$ and is assumed to be a subset of two dimensional Euclidean space $\mathbf{S} \subset \mathbb{R}^2$. The vector $s \in \mathbf{S}$ describes a point in $\mathbf{S}$.

Let X denote the state of the environment with respect to target presence. In general, X is a discrete time random variable that takes values on alphabet $\mathcal{X}$. For the stationary target task that we consider, X is constant over time. Y is a sensor observation (also a random variable) of a portion of the environment that takes values on an alphabet $\mathcal{Y}$. We assume that environmental sensor measurements occur at discrete times and are indexed by the variable $t = 1, 2, 3 \ldots$.

Let Z denote the state of the environment with respect to hazard presence, where Z is a discrete time random variable that takes values on alphabet $\mathcal{Z}$. Let Q be an observation of a failure event at a location in the environment, where Q takes values on alphabet $\mathcal{Q}$. Even though we cannot directly observe agent failures, we can still reason about the probability that they occur; thus, defining Q is useful. We assume the environment is discretized such that paths are broken to a finite number of edges between nodes. The traversal of path segments from node to node is modeled (for the purpose of hazard inference) by a global discrete time counter that uses the variable $\tau = 1, 2, 3 \ldots$. Target sensor measurements may happen independently of path segment traversals such that $\tau \neq t$, in general. We assume the ability to track both τ and t (independently).

We are provided a set W of k fixed uplink sites, $W = \{w_1, \ldots, w_k\}$ where agents can upload the data they collect, e.g., for future use. A path ζ is a mapping from the interval $[0, 1]$ to the state space $\mathbf{S}$. We will assume that paths are piecewise continuous curves that start and end at uplink sites. Formally, $\zeta : [0, 1] \rightarrow \mathbf{S}$ such that $\zeta(0) = s_{\text{start}} = w_a$ and $\zeta(1) = s_{\text{goal}} = w_b$ for $w_a, w_b \in W$. We allow both $w_a = w_b$ and $w_a \neq w_b$. With an abuse of notation (made to improve the overall clarity of our presentation) we overload the symbol ζ to additionally represent the set of points contained in the path that it defines.

The robot is assumed to take sensor measurements regarding target presence as it travels along the path. In this paper we consider the discrete case where one observation is made at each node in the path. Given this assumption, and assuming that t measurements have already been taken *before* an agent starts moving along its path, then the successful completion of a path provides an ordered finite set of sensor observations $\{y_{t+1}, \ldots, y_{t+\ell}\}$, where y_k is taken at position $s_k \in \zeta$ and $t + 1 \leq k \leq t + \ell$, where ℓ is the number of sensor readings taken along the path ζ.

Given a hazardous environment (as well as a nominal probability of agent failure), the successful traversal of the path is itself a random event that depends

on both the path taken, and the hazards in the environment. Let $\theta_{\zeta,\text{alive}}$ and $\theta_{\zeta,\text{dead}}$ denote the complementary events that the robot survives the path ζ or does not, respectively. Let Θ_ζ be the random variable associated with survival of a path ζ. In general, the probability of these events is defined by a functional that accounts for motion along the path, and environmental hazards. $\mathbb{P}(\Theta_\zeta = \theta_{\zeta,\text{alive}}|Z) = f(\zeta, Z)$ and $\mathbb{P}(\Theta_\zeta = \theta_{\zeta,\text{dead}}|Z) = 1 - \mathbb{P}(\Theta_\zeta = \theta_{\zeta,\text{alive}}|Z)$. The particular form of f depends on the way that the environment is modeled. In our experiments we assume that hazards destroy agents transiting their map cells with probability $p_{\text{kill}} \in (0, 1]$, and that agents also malfunction with probability $p_{\text{malfunc}} \in [0, 1)$ in each cell.

4 Bayesian Belief Updates and Expected Information Gain

The recursive Bayesian update for the path-based sensor requires a discrete time model. This model supports continuous paths, so long as the path can be partitioned into a finite number ℓ of path segments in order to reason about the location of the map cell in which an agent may have been destroyed. In a positive path-based sensor "observation" of a hazard the observing agent is destroyed, which prevents it from relaying direct knowledge about the hazard to other agents. However, a belief update is possible by considering separately each possibility (the finite set of mutually exclusive events that the agent was destroyed while traveling along each segment $k \in [1, \ell]$), and then combining the resulting ℓ separate belief maps weighted by the relative likelihood of each occurring. Whenever the agent survives the path we can directly update our belief map based on negative hazard observation occurring along each of the ℓ segments in the path.

4.1 Target Updates (Assuming a Standard Sensor)

Let X_0 denote the prior belief defined over $\mathbf{S}$ that each point $s \in \mathbf{S}$ contains a target. For notational convenience, we increment the time index t based on the number of successfully communicated sensor measurements, i.e., t ordered sensor observations have been delivered to the uplink points by time t. Given sensor measurements $y_1, \ldots, y_t$ (which may have been taken across multiple paths of varying lengths) and X_0, an iterative Bayesian update can be used to compute $\mathbb{P}(X_t|y_1, \ldots, y_t)$, the posterior probability of X given the t sensor readings delivered to the uplink points by time t.

$$\mathbb{P}(X_t|Y_1 = y_1, \ldots, Y_t = y_t) = \frac{\mathbb{P}(Y_t=y_t|X_{t-1})\mathbb{P}(X_{t-1}|Y_1=y_1,\ldots,Y_{t-1}=y_{t-1})}{\mathbb{P}(Y_t=y_t|Y_1=y_1,\ldots,Y_{t-1}=y_{t-1})} \quad (1)$$

As is standard practice, the denominator need not be explicitly calculated; rather, we calculate the numerators of Eq. 1 for all events $X_t = x \in \mathcal{X}$ and then normalize so that they sum to 1.

The information entropy of X_t is denoted $H(X_t)$ and defined:

$$H(X_t) = -\int_{x \in \mathcal{X}} \mathbb{P}(X_t) \log \mathbb{P}(X_t)\, dx$$

and provides a measure of the unpredictability of X_t. As entropy increases, X_t is essentially "worse" at being able to predict the presence or absence of a target (in other words, its values are closer to a uniformly random process).

The conditional information entropy $H(X_{t+1}|Y_{t+1})$ is the updated entropy of the environmental state X (w.r.t. target presence) given a new observation Y_{t+1}, *averaged over all possible values that* Y_{t+1} *may take*. The difference between the entropy $H(X_t)$ and the conditional entropy $H(X_{t+1}|Y_{t+1})$ is called *mutual information*, defined $I(X_t; Y_{t+1}) = H(X_t) - H(X_{t+1}|Y_{t+1})$. Mutual information quantifies the *expected* reduction in the unpredictability of our estimation of X given the new measurement Y_{t+1}.

It is useful to calculate the mutual information of a new (target sensor) measurement Y_{t+1} before it is taken, so that we may compare the expected benefits of sampling various locations. The mutual information of a new observation (assuming it is delivered to a communication point) is calculated:

$$I(X_t; Y_{t+1}) = \int_{y \in \mathcal{Y}} \int_{x \in \mathcal{X}} \mathbb{P}(Y_{t+1} = y, X_t = x) \log \left(\frac{\mathbb{P}(Y_{t+1} = y, X_t = x)}{\mathbb{P}(Y_{t+1} = y)\, \mathbb{P}(X_t = x)} \right) dxdy$$

where $\mathbb{P}(Y_{t+1} = y, X_t = x) = \mathbb{P}(Y_{t+1} = y | X_t = x)\, \mathbb{P}(X_t = x)$.

We want to plan paths that gather as much mutual information along the path as possible, given fuel constraints and other goals. Another goal, for example, is to also gather information about hazard locations (hazards are discussed in the next subsection). Given a path ζ that enables sensor observations $y_{t+1}, \ldots, y_{t+\ell}$ if and only if it is completed successfully, the expected cumulative information gained along that path (given all measurements so far) is calculated:

$$I(X_t; Y_{t+1}, \ldots, Y_{t+\ell}) = \textstyle\sum_{k=t+1}^{t+\ell} I(X_{k-1}; Y_k | Y_{t+1}, \ldots, Y_{k-1})$$

Where the notation $I(A; B|C)$ denotes the conditional mutual information of A and B, integrated over all possible outcomes in the event space of C (and weighted by their relative likelihood). That is, $I(A; B|C) = \mathbb{E}_C\left(I(A; B)|C\right)$.

In the most general case (in which targets at any locations in the environment may affect sensor readings at any other location) the calculation of $I(X_t; Y_{t+1}, \ldots, Y_{t+\ell})$ can become intractable because the number of terms involved in the computation of the inner $I(X_{k-1}; Y_k | Y_{t+1}, \ldots, Y_{k-1})$ scales according to $|\mathcal{Y}|^k$. However, this complexity can be reduced, e.g., to a small constant, by assuming that each target only affect sensor observations in its own local neighborhood.

From the point-of-view of the planning system, no information about target locations is actually gained until the robot reaches an uplink point. Hence, *no information about* ***targets*** *is gathered in the event that the robot is destroyed*

along its path. Consequently, the expected mutual information along a particular path (assuming the robot may or may not be destroyed along that path) is:

$$I(X_t; Y_{t+1}, \ldots, Y_{t+\ell} | \Theta_\zeta) = \mathbb{P}\left(\Theta_\zeta = \theta_{\zeta,\text{alive}}\right) I(X_t; Y_{t+1}, \ldots, Y_{t+\ell}).$$

4.2 Hazard Updates (Assuming a Path-Based Sensor)

Environmental hazards may prohibit agents from reaching their intended destinations. Thus, *we can update our belief about environmental hazards Z by observing whether or not agents reach their destinations.* Both events $\Theta_\zeta = \theta_{\zeta,\text{alive}}$ and $\Theta_\zeta = \theta_{\zeta,\text{dead}}$ can be used to perform an iterative Bayesian update of Z based on ζ. However, the iterative updates to Z based on Θ_ζ take different forms depending on if $\Theta_\zeta = \theta_{\zeta,\text{alive}}$ or $\Theta_\zeta = \theta_{\zeta,\text{dead}}$.

We begin by noting that *if* we had access to data of hazard observations at all cells along the path, then a straightforward belief update is as follows:

$$\mathbb{P}\left(Z_{\tau+j} | Z_\tau, Q_{\tau+1} = q_{\tau+1}, \ldots, Q_{\tau+j-1} = q_{\tau+j-1}\right) = \frac{\mathbb{P}\left(Q_{\tau+j} = q_{\tau+j} | Z_{\tau+j-1}\right) \mathbb{P}\left(Z_{\tau+j-1} | Z_\tau, Q_{\tau+1} = q_{\tau+1}, \ldots, Q_{\tau+j-1} = q_{\tau+j-1}\right)}{\mathbb{P}\left(Q_\tau = q_\tau | Z_\tau, Q_{\tau+1} = q_{\tau+1}, \ldots, Q_{\tau+j-1} = q_{\tau+j-1}\right)} \tag{2}$$

Next, we observe that whenever an agent survives we *do* have direct access to all "observations" of hazards along the path, and they are $Q_j = q_j = 0$ by construction (since the agent survived). Formally, $\Theta_\zeta = \theta_{\zeta,\text{alive}} \iff q_{\tau+1} = 0, \ldots, q_{\tau+l} = 0$. Thus, we simply perform the standard update:

$$\mathbb{P}\left(Z_{\tau+j} | \Theta_\zeta = \theta_{\zeta,\text{alive}}\right) = \mathbb{P}\left(Z_{\tau+l} | Z_\tau, Q_{\tau+1} = 0, \ldots, Q_{\tau+l} = 0\right)$$

which can be computed iteratively, for each $j = 1, \ldots, l$ as follows:

$$\mathbb{P}\left(Z_{\tau+j} | Z_\tau, Q_{\tau+1} = 0, \ldots, Q_{\tau+j-1} = 0\right) = \frac{\mathbb{P}(Q_{\tau+j}=0 | Z_{\tau+j-1}) \mathbb{P}(Z_{\tau+j-1} | Z_\tau, Q_{\tau+1}=0, \ldots, Q_{\tau+j-1}=0)}{\mathbb{P}(Q_\tau = 0 | Z_\tau, Q_{\tau+1}=0, \ldots, Q_{\tau+j-1}=0)}$$

In contrast, when an agent does not survive ($\Theta_\zeta = \theta_{\zeta,\text{dead}}$) the recursive Bayesian update of Z must take a different form. Given a path with l segments, with the first segment starting at time τ, the event $Q_{\tau+j} = 1$ is equivalent to the statement "the agent was killed along the j-th segment of the path".

Given $\Theta_\zeta = \theta_{\zeta,\text{dead}}$, we know that the agent was killed *somewhere* along ζ, but we do not know where. However, we can integrate over all l possibilities, i.e., considering each possibility that the robot was killed on path segment j for all j such that $1 \leq j \leq l$, and then summing these results weighted by the relative probability of each (given our current hazard beliefs).

It is convenient to use the metaphor of a multiverse. We simultaneously assume the existence of j different universes, such that in the j-th universe the agent was killed along the j-th path segment. Assuming we are in a particular j-th universe, we can calculate the iterative Bayesian update to Z by applying Eq. 2 exactly j times, assuming that on the k-th application:

$$Q_{\tau+k} = q_{\tau+k} = \begin{cases} 0 & \text{if } k < j \\ 1 & \text{if } k = j \end{cases}$$

and where no observations are made for $k > j$ in the j-th universe. Let $Z^j_{\tau+l}$ denote the version of $Z_{\tau+l}$ that is calculated in the j-th universe.

The final overall update to the "real" $Z_{\tau+l}$ is the expected value of $Z_{\tau+l}$ in the multiverse, found by combining all $Z_{\tau+l}$ weighted by $\mathbb{P}\left(Q_{\tau+j} = 1 | Z_\tau, \Theta_\zeta = \theta_{\zeta,\text{dead}}\right)$, the probability of being in the j-th universe.

$$Z_{\tau+l} = \sum_{j=1}^{l} \mathbb{P}\left(Q_{\tau+j} = 1 | Z_\tau, \Theta_\zeta = \theta_{\zeta,\text{dead}}\right) Z^j_{\tau+l} \tag{3}$$

The quantity $\mathbb{P}\left(Q_{\tau+j} = 1 | Z_\tau, \Theta_\zeta = \theta_{\zeta,\text{dead}}\right)$ can be obtained by calculating the probability that the agent survives to the j-th path segment given Z_τ and is then destroyed there, and then normalizing such that the probabilities of all l possibilities sum to 1.

$$\mathbb{P}\left(Q_{\tau+j} = 1, Z_\tau, \Theta_\zeta = \theta_{\zeta,\text{dead}}\right) = \frac{\mathbb{P}\left(Q_{\tau+j} = 1, Z_\tau\right) \prod_{k=1}^{j-1} \mathbb{P}\left(Q_{\tau+k} = 0, Z_\tau\right)}{\sum_{j=1}^{l} \mathbb{P}\left(Q_{\tau+j} = 1, Z_\tau\right) \prod_{k=1}^{j-1} \mathbb{P}\left(Q_{\tau+k} = 0, Z_\tau\right)}$$

where $\mathbb{P}\left(Q_{\tau+k} = q, Z_\tau\right) = \int_{z \in \mathcal{Z}} \mathbb{P}\left(Q_{\tau+k} = q | Z_\tau = z_\tau\right) \mathbb{P}\left(Z_\tau = z_\tau\right) dz$ for $q \in \{0, 1\}$.

The expected decrease in entropy about hazard locations gained from sending an agent along path ζ can be calculated by first calculating the conditional decrease in entropy assuming either possibility of $\Theta_\zeta = \theta_{\zeta,\text{alive}}$ and $\Theta_\zeta = \theta_{\zeta,\text{dead}}$ (i.e, independently), and then combining the results weighted by the probability of each event given Z_τ.

Let $Z^{\theta_{\zeta,\text{alive}}}_{\tau+l}$ be the value of $Z_{\tau+l}$ that results if the agent survives the path (as calculated according to Eq. 2). Similarly, let $Z^{\theta_{\zeta,\text{dead}}}_{\tau+l}$ be the result if the agent does not survive (as calculated by Eq. 3).

The mutual information regarding hazards (the expected information gained from a path-based sensor observation) is given by the expected reduction in entropy: $I(Z_\tau; \Theta_\zeta) = H(Z_\tau) - H(Z_{\tau+l} | \Theta_\zeta, Z_\tau)$, where

$$H(Z_{\tau+l} | \Theta_\zeta, Z_\tau) = \int_\theta \mathbb{P}\left(\Theta_\zeta = \theta | Z_\tau\right) H(Z_{\tau+l} | \Theta_\zeta = \theta, Z_\tau) d\theta$$

is the conditional entropy, and $\mathbb{P}\left(\Theta_\zeta = \theta_{\zeta,\text{alive}} | Z_\tau\right) = \prod_{j=1}^{l} \mathbb{P}\left(Q_{\tau+j} = 0, Z_\tau\right)$ and $\mathbb{P}\left(\Theta_\zeta = \theta_{\zeta,\text{dead}} | Z_\tau\right) = 1 - \mathbb{P}\left(\Theta_\zeta = \theta_{\zeta,\text{alive}} | Z_\tau\right)$ and $H(Z_{\tau+l} | \Theta_\zeta = \theta_{\zeta,\text{alive}}, Z_\tau) = Z^{\theta_{\zeta,\text{alive}}}_{\tau+l}$ and $H(Z_{\tau+l} | \Theta_\zeta = \theta_{\zeta,\text{dead}}, Z_\tau) = Z^{\theta_{\zeta,\text{dead}}}_{\tau+l}$.

5 Formal Problem Definitions

Problem Definition. *Mutual information path planning for targets and hazards without communication*:

Given a search space $\mathbf{S}$, and a set of uplink points $\{w_1, \ldots, w_k\} = W \subset \mathbf{S}$, and assuming an agent can start at any $w_{start} \in \mathbf{S}$ and end at any $w_{goal} \in \mathbf{S}$, and assuming an agent has a noisy target sensor that provides observations Y

about targets X. Find the path ζ^* that maximizes the expected information gain about both targets X and hazards Z:

$$\zeta^* = \arg\max_{\zeta} c_X I(X_t; Y_{t+1}, \ldots, Y_{t+\ell} | \Theta_\zeta) + c_Z I(Z_\tau; \Theta_\zeta)$$

where $c_X, c_Z \in [0,1]$ are weights that represent user preference for either type of information, and where for all ζ it is true that $\zeta : [0,1] \rightarrow \mathbf{S}$ s.t. $\zeta(0) = w_{start}$ and $\zeta(1) = w_{goal}$, subject to distance constraints $\|\zeta\| < \ell$, and Θ_ζ is the space of observations about whether or not the agent successfully completes the path.

Problem Definition. *Iterative mutual information path planning for targets and hazards without communication*:

Repeatedly solve the mutual information path planning for targets and hazards problem to continually refine our belief about targets X and hazards Z given Y and Θ_ζ, respectively; assuming we are able to replace agents that are lost (once their failure to appear at their destinations as been observed).

6 Algorithms

The environment is modeled as discrete map $\mathbf{M}$ of non-overlapping cells $M_i \subset \mathbf{M}$, where $1 \leq i \leq m$ and $M_i \cap M_j = \emptyset$ for $i \neq j$. Target and adversary effects are assumed local to the map cells containing those targets and adversaries, respectively. These assumptions are useful in practice because they reduce computational complexity. To simplify our presentation the same map $\mathbf{M} = \bigcup_{i \in [1,m]} M_i$ is used to reason about both targets and hazard. Because target and hazard effects are local to each cell, our beliefs about targets (and hazards) are stored in arrays $\mathbf{X}$ (and $\mathbf{Z}$), where $\mathbf{X}[i]$ (and $\mathbf{Z}[i]$) are our current belief that map cell M_i contains a target[3] (respectively, a hazard). Connectivity information is stored in a graph $G_\mathbf{S} = (V_\mathbf{S}, E_\mathbf{S})$. Each map cell M_i has a corresponding node $v_i \in V_\mathbf{S}$, and an edge $(v_i, v_j) \in E_\mathbf{S}$ indicates it is possible to move directly between map cells M_i and M_j. Self transitions $(v_i, v_i) \in E_\mathbf{S}$ are allowed, but can be removed in cases where agents must remain in motion.

Mutual information is sub-modular—there are diminishing returns for visiting the same cell again and again. Therefore, we plan in $G_{\mathbf{S}\times\mathbb{T}} = (V_{\mathbf{S}\times\mathbb{T}}, E_{\mathbf{S}\times\mathbb{T}})$ the space-time extension of $G_\mathbf{S}$, to track cell visit counts along a path.

[3] In the most general discrete formulation of the ideas presented in Sects. 3, 4 and 5, target existence across all cells in the map is represented by a single random variable X that takes one of the 2^m different possible values x (depending on which cells contain targets and which do not). The set of all 2^m possibilities forms the alphabet $\mathcal{X}$. However, if each target only affects target sensor readings in its own cell, then the resulting independence between cells allows us to consider each of the m dimensions of X separately. In other words, we can consider X as a joint event over a collection of independent random variables $X_1, \ldots, X_m$ because $\mathbb{P}(X = x) = \prod_{i=1}^m \mathbb{P}(X_i = x_i)$. We store our current estimate of $\mathbb{P}(X_i = x_i)$ in $\mathbf{X}[i]$.

Agents have fuel for ℓ moves, so $G_{\mathbf{S}\times\mathbb{T}}$ is created by placing a "clone" $V_{\mathbf{S},t} \equiv V_{\mathbf{S}}$ at each of the $0 \leq t \leq \ell+1$ time steps that must be considered, i.e., $V_{\mathbf{S}\times\mathbb{T}} = V_{\mathbf{S},0} \cup \ldots \cup V_{\mathbf{S},\ell}$. Edges in $E_{\mathbf{S}\times\mathbb{T}}$ move forward in time, and exist according to the following rule: $(v_i, v_j) \in E_{\mathbf{S}} \implies (\hat{v}_{i,t-1}, \hat{v}_{j,t}) \in E_{\mathbf{S}\times\mathbb{T}}$ for all $t \in [1, \ell]$. A valid path ζ_{valid} is a sequence of edges that starts at some uplink site $w_{start} = \hat{v}_{j,0}$ at time $t = 0$ and moves from node to node along edges in space-time until reaching a (goal) uplink site $w_{goal} = \hat{v}_{j,\ell}$ at time $t = \ell$.

If β is a belief that one of two complementary events has occurred, its entropy is calculated: $H(\beta) = -(\beta \log(\beta) + (1-\beta)\log(1-\beta))$. Given our assumption of cell independence, total entropy regarding targets is $H(\mathbf{X}) = \sum_{i=1}^{m} H(\mathbf{X}[i])$ and total entropy regarding hazards is $H(\mathbf{Z}) = \sum_{i=1}^{m} H(\mathbf{Z}[i])$. Let $p_\zeta^{\text{alive}} \equiv \mathbb{P}(\theta_{\zeta,\text{alive}})$ and $p_\zeta^{\text{dead}} \equiv \mathbb{P}(\theta_{\zeta,\text{dead}})$.

The outer loop of the iterative planning approach appears in Algorithm 1, and the algorithm used to plan each path appears in Algorithm 3. We track the expected entropy that results from attempting paths using the arrays $\mathbf{X}_\zeta$ and $\mathbf{Z}_\zeta$ (for targets and hazards, respectively). $\mathbf{X}_\zeta \equiv \mathbb{E}_\zeta(\mathbf{X})$, where $\mathbb{E}_\zeta(\mathbf{X}) = p_\zeta^{\text{alive}}\mathbb{E}_\zeta(\mathbf{X}|\theta_{\zeta,\text{alive}}) + p_\zeta^{\text{dead}}\mathbb{E}_\zeta(\mathbf{X}|\theta_{\zeta,\text{dead}})$. All sensor readings about targets are lost if the agent is killed, thus $\mathbb{E}_\zeta(\mathbf{X}|\theta_{\zeta,\text{dead}}) = 0$ and so $\mathbb{E}_\zeta(\mathbf{X}) = p_\zeta^{\text{alive}}\mathbb{E}_\zeta(\mathbf{X}|\theta_{\zeta,\text{alive}})$. In contrast, information about hazard existence is gained both if the agent survives or if the agent is destroyed (though different amounts in either case). We use the vectors $\mathbf{Z}_\zeta^{\text{alive}} \equiv \mathbb{E}_\zeta(\mathbf{Z}|\theta_{\zeta,\text{alive}})$ and $\mathbf{Z}_\zeta^{\text{dead}} \equiv \mathbb{E}_\zeta(\mathbf{Z}|\theta_{\zeta,\text{dead}})$ to track the conditional expectations of $\mathbf{Z}_\zeta$ that will result if the agent survives or is killed along path ζ. This allows us to compute $\mathbf{Z}_\zeta \equiv \mathbb{E}_\zeta(\mathbf{Z}) = p_\zeta^{\text{alive}}\mathbf{Z}_\zeta^{\text{alive}} + p_\zeta^{\text{dead}}\mathbf{Z}_\zeta^{\text{dead}}$.

Algorithm 1. Iterative information path planning for targets and hazards

for $r = 1, 2, \ldots$ **do**
  $\zeta =$ `calculatePath`$(\mathbf{X}, \mathbf{Z})$
  Robot r attempts path ζ
  if $\theta_{\zeta,\text{alive}}$ **then**
    $\mathbf{X} \leftarrow$ `BayesianCellUpdates`$(\mathbf{X}, \mathbf{Y}_\zeta)$
    $\mathbf{Z} \leftarrow$ `BayesianCellUpdates`$(\mathbf{Z}, [0, \ldots, 0])$
  else
    $\mathbf{Z} \leftarrow$ `KilledOnPathUpdate`$(\mathbf{Z})$

Algorithm 2. `KilledOnPathUpdate`$(\mathbf{Z}, \zeta)$

$p_1^{survivedTo} \leftarrow 1$
for $k \leftarrow 1, \ldots, \ell$ **do**
  $i \leftarrow$ index of cell in which k-th observation was made
  $p_k^{killedInGivenAt} \leftarrow (p_{\text{kill}} + p_{\text{malfunc}}(1 - p_{\text{kill}}))\mathbf{Z}[i] + p_{\text{malfunc}}(1 - \mathbf{Z}[i])$
  $p_{k+1}^{survivedTo} \leftarrow p_k^{survivedTo}(1 - p_k^{killedInGivenAt})$
  $\mathbf{Z}_k \leftarrow \mathbf{Z}$
  $\mathbf{Z}_k \leftarrow$ `BayesianCellUpdates`$(\mathbf{Z}_k, [\mathbf{0}_{1:k-1}, 1])$
$p_\zeta^{\text{dead}} \leftarrow \sum_{k=1}^{\ell} p_k^{survivedTo}$
$\mathbf{Z} \leftarrow \sum_{k=1}^{\ell} \frac{p_k^{survivedTo}}{p_\zeta^{\text{dead}}}\mathbf{Z}_k$
return $(\mathbf{Z}, (1 - p_\zeta^{\text{dead}}))$

Algorithm 3. `calculatePath`$(\mathbf{X}, \mathbf{Z})$

for all uplink points $w \in W_{\text{goal}}$ **do**
  $\zeta_w \leftarrow \emptyset$
  `InsertFIFOQueue`(w)
while $\hat{v}_j \leftarrow$ `PopFIFOQueue` **do**
  $h_{\hat{v}_j} = -\infty$
  for all $(\hat{v}_i, \hat{v}_j) \in E_{\mathbf{S}\times\mathbb{T}}$ **do**
    $\zeta \leftarrow (\hat{v}_i, \hat{v}_j) + \zeta_{\hat{v}_j}$
    $\hat{h}_{live}^{\mathcal{X}} \leftarrow \int_{x\in\mathcal{X}} H(\mathbf{X}_{live})dx$
    $(\mathbf{Z}_{live}, p_\zeta^{\text{alive}}) \leftarrow$ `KilledOnPathUpdate`$(\mathbf{Z}, \zeta)$
    $\mathbf{Z}_{killed} \leftarrow$ `BayesianCellUpdates`$(\mathbf{Z}, [0, \ldots, 0])$
    $\hat{h}_{this} \leftarrow c_Z(p_\zeta^{\text{alive}}H(\mathbf{Z}_{live}) + (1 - p_\zeta^{\text{alive}})H(\mathbf{Z}_{killed})) + c_X p_\zeta^{\text{alive}}\hat{h}_{live}^{\mathcal{X}}$
    if $\hat{h}_{this} > h_{\hat{v}_j}$ **then**
      $\zeta_{\hat{v}_i} \leftarrow \zeta$
      $h_{\hat{v}_j} \leftarrow h_{this}$
$w_{start} \leftarrow \arg\min_{w\in W_{\text{goal}}} \hat{v}_j$
return $\zeta_{w_{start}}$

Algorithm 4. `BayesianCellUpdates`$(\mathbf{B}, \bar{\beta})$

for $k = 0, \ldots, \ell$ **do**
  $i \leftarrow$ index of cell in which k-th observation was made
  $\mathbf{B}[i] \leftarrow \mathbb{P}(B_i \mid \mathbf{B}[i], \bar{\beta}[k])$
return $\mathbf{B}$

Cell-wise target and hazard observations made along path ζ are stored in the vectors $\mathbf{Y}_\zeta$ and $\mathbf{Q}_\zeta$, where $\mathbf{Y}_\zeta[k]$ and $\mathbf{Q}_\zeta[k]$ are the observations made in the k-th cell along the path. $\mathbf{Y}_\zeta[k] \in \{1, 0\}$ where 1 denotes that a sensor reading was positive and 0 denotes that a sensor reading was negative. Similarily, $\mathbf{Q}_\zeta[k] \in \{1, 0\}$, where 1 represents a hazard observation and 1 denotes a negative (cell-wise) reading. Even though it is impossible for an agent to directly report a hazard observation, $\mathbf{Q}_\zeta$ is used for two things: first, $\mathbf{Q}_\zeta[k] = 0$ for all $k \in [0, \ell]$ when an agent survives. Second, the algorithm tracks a different version of $\mathbf{Q}_\zeta$ for each member of the set of possible events, when reasoning about the relative probabilities of survival to different places along a path.

Algorithm 4 shows the recursive Bayesian update that is used for the belief vector $\bar{\beta}$ given the observation vector B_i. Depending on context $\bar{\beta}$ may represent $\mathbf{Y}_\zeta$ or $\mathbf{Q}_\zeta$, and $\mathbf{B}$ may represent X and Z. Line 3 performs the recursive update yielding the posterior probability B_i regarding existence in the i-th map cell.

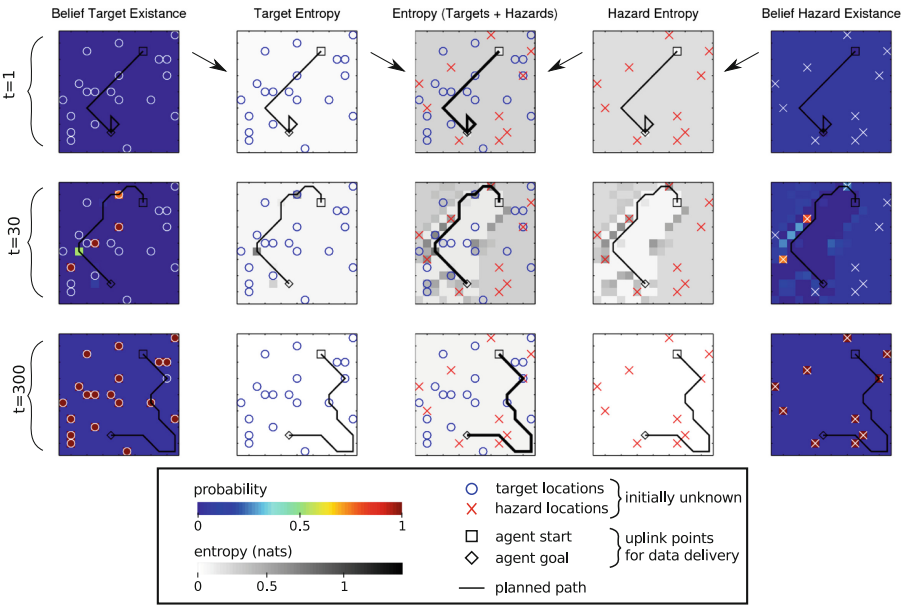

Fig. 2. Top to bottom: adversary and target existence beliefs and their entropies over time. Left two columns relate to targets. Right two columns to hazards. Center column is the sum of target and hazard entropies. Paths are calculated to maximize the (user weighted) reduction in entropy over the set of complementary events that the agent survives or is destroyed, weighted by their probability. At $t = 1$ the prior beliefs of hazard and target existence are 0.01 and 0.05 in each map cell. At $t = 300$ the agent has accurate beliefs of all 10 hazards and 19 of 20 targets, and is working (at high risk of being destroyed) to gather target existence information in a cell that contains a known hazard. In this trial hazards have a 50% kill rate.

7 Experiments

We use Monte Carlo trials in simulation to evaluate the performance of the algorithm presented in Sect. 6, and compare it to other approaches. For the experiments presented in this section, the environment is represented by a 15×15 grid map. Movement is defined by a 9-grid of connectivity (8-grid neighbors plus self transitions) where each move takes the agent one time-step further in time. The agent has enough endurance to make 25 moves. Agent malfunction rate is $p_{\text{malfunc}} = .01$ per time step (thus, on average agents arbitrarily malfunction in 1/4 of all forays of length 25). In each trial the start and goal uplink points are placed uniformly at random. 10 non-start/goal locations are picked uniformly at random (no replacement) and populated with hazards. This is repeated for 20 non-start/goal locations that are populated with targets.

We test our algorithm using three different objectives: weighting information from targets and hazards equally $c_X, c_Z = \{1, 1\}$; gathering only target information $c_X, c_Z = \{1, 0\}$; and gathering only hazard information $c_X, c_Z = \{0, 1\}$. We compare to three other ideas: (1) 1-step look ahead information surfing[4]; (2) a Markov random walk; and (3) planning paths to gather target information while ignoring hazards altogether (by not accounting for the probability of being destroyed when evaluating the expected information gain, and assuming a $c_X, c_Z = \{1, 0\}$ objective).

Our method and information surfing both track and update target and hazard beliefs, and use the probability of hazard existence to weight the expected information that will be gained about targets and/or hazards. The path of the random walk is calculated before the agent departs such that the resulting-path sensor can be used to infer hazard presence based on whether or not it survives. In all methods agent movement is only allowed in directions from which the agent can still reach the goal given its fuel reserves.

Figure 2 shows examples paths for an experiment in the same environment with adversary lethality of 0.5. To generate performance statistics, each method is tested on (the same) 30 randomly generated configurations, and across six different adversary lethality levels (0.01, 0.2, 0.4, 0.6, 0.8, and 0.99). Due to space

[4] In "information surfing" the path is greedily computed—the path is initially unknown when that agent leaves the start and then the path is computed on-the-fly. Hazard existence belief is tracked and used to determine the expected information that will be gained about targets. In practice, the destruction of an agent eliminates direct knowledge of the path taken by the agent. While it may be possible to integrate over all possible paths the agent could have taken to obtain a valid update, this computation is at least as hard as the algorithm we present for planning the optimal path. Instead, for the purposes of comparison we choose to be overly generous to "information surfing" and (unrealistically) assume that if the agent is destroyed, then we still know the path that it would have taken to the goal had it not been destroyed. Using this path to refine hazard beliefs (by calculating the relative likelihood the agent was destroyed on each segment) provides a performance bound such that the results for "information surfing" are better than what is expected in practice.

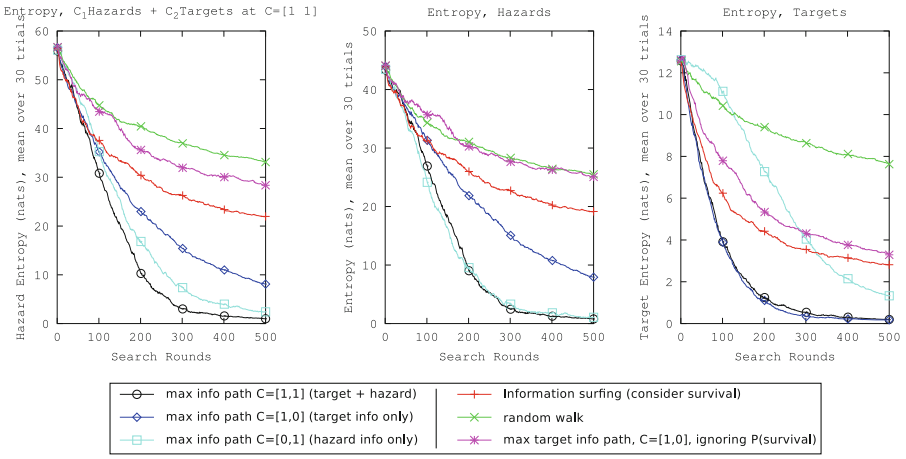

Fig. 3. Hazard and target entropies vs. search round (mean over 30 random trials) when hazards have a 60% kill ratio. Left: an equally weighted ($C = [11]$) combination of hazard and target entropy. Center: target entropy. Right: hazard entropy.

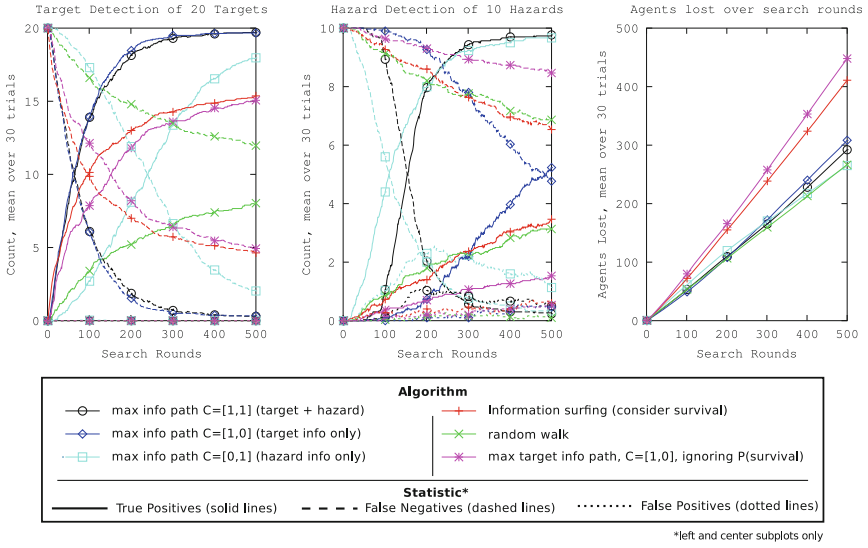

Fig. 4. Left: target statistics. Center: hazard statistics. Right: total agent losses over time. Statistics include true/false positives and false negatives of target locations. Likelihood $\geq .95$ is defined as a positive detection. All plots show mean results over 30 random trials. In these experiments hazards have a 60% kill success rate. (Agents are expendable such that any costs associated with their losses are negligible compared to the information that is gained from their loss).

limitations we only present results for the 0.6 case. In Figs. 3 and 4 and results for other cases appear in supplementary material available on the first author's website. Regarding the calculation of true/false positives, if target existence belief is $\geq$0.95 in a cell, then we declare that we think there is a target in that cell. Likewise, for hazards. The relative performance of different methods is similar across hazard lethality rates, though increasing lethality rates make the detection of hazards easier and the detection of targets harder for all methods.

The method we present consistently outperforms the other methods on the objective it seeks to maximize. However, in the case $c_X, c_Z = \{0, 1\}$ for which our method seeks information about only *hazards* and not targets, then all of the comparison methods initially have more accurate beliefs regarding *target* existence (however they also all eventually fall behind our method at later iterations). This makes sense given that our algorithm is completely ignoring target information in its mission objective in that particular case. In general, "Information surfing" performs better than the random walk, but not as well as full information based path planning. Information based target search that completely ignores the possibly of being destroyed by a hazard has the worst performance of all methods tested. This is because target beliefs remain unchanged in the event that agents are destroyed—if hazards are ignored then subsequent agents will continue to attempt the same dangerous path until it is successfully completed.

8 Summary and Conclusion

An agent's path can be used as a binary sensor (to detect the occurrence of at least one event along that path), and we show how to compute recursive Bayesian updates given such path-based sensor observations. In hazardous environments where communication with an agent is impossible until it physically returns, this allows the existence of lethal hazards to be inferred based on whether or not agents survive forays along paths.

By calculating the expected information that will be gained along different paths, we are able to maximize the information about hazards that is expected to be gained along each foray. This idea is combined with standard Bayesian target search to provide a family of algorithms for solving the problem of iterative path planing for target search in hazardous environments without communication.

In Monte Carlo simulations presented in Sect. 7 we find that the algorithms perform favorably vs. three related ideas including: (1) "information surfing" which has been shown to work well for the related problem of target search in a hazardous environment *with* communication; (2) performing informative path planning for targets while ignoring hazards; and (3) a Markov random walk that is computed before the agent leaves. When the mission objective is set to maximize the information that is collected regarding targets, hazards, or a weighted combination of both, then the detection of targets, hazards, or both are respectively maximized (based on 95% belief defining a positive observation).

The Bayesian belief updates for a path-based sensor can be useful even if agents do not use hazard information in their mission objectives. For example,

using random walks will eventually result in accurate beliefs of hazard existence (at least, almost surely in the limit). Thus, even if agents perform a variety of other missions in the environment, we can still use observations of their survival vs. destruction to refine our beliefs of hazard locations.

Acknowledgments. This work was performed at the Naval Research Laboratory (NRL) and funded by Office of Naval Research (ONR) grant N0001416WX01271. The views, positions and conclusions expressed herein reflect only the authors' opinions and expressly do not reflect those of ONR, nor those of NRL.

References

1. Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics. Auton. Robot. **31**(4), 299–316 (2011)
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Willey-Interscience, Hoboken (2006)
3. Dames, P., Schwager, M., Kumar, V., Rus, D.: A decentralized control policy for adaptive information gathering in hazardous environments. In: IEEE Conference on Decision and Control, pp. 2807–2813, December 2012
4. Dames, P., Kumar, V.: Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. IEEE Trans. Autom. Sci. Eng. **12**(3), 850–864 (2015)
5. Dames, P.M., Schwager, M., Rus, D., Kumar, V.: Active magnetic anomaly detection using multiple micro aerial vehicles. IEEE Robot. Autom. Lett. **1**(1), 153–160 (2016)
6. Flint, M., Polycarpou, M., Fernandez-Gaucherand, E.: Cooperative control for multiple autonomous UAV's searching for targets. In: IEEE Conference on Decision and Control, vol. 3, pp. 2823–2828, December 2002
7. Flint, M., Fernández-Gaucherand, E., Polycarpou, M.: Cooperative control for UAV's searching risky environments for targets. In: IEEE Conference on Decision and Control, vol. 4, pp. 3567–3572. IEEE (2003)
8. Hollinger, G.A., Yerramalli, S., Singh, S., Mitra, U., Sukhatme, G.S.: Distributed data fusion for multirobot search. Trans. Robot. **31**(1), 55–66 (2015)
9. Julian, B.J., Angermann, M., Schwager, M., Rus, D.: Distributed robotic sensor networks: an information-theoretic approach. Int. J. Robot. Res. **31**(10), 1134–1154 (2012)
10. Lyu, Y., Chen, Y., Balkcom, D.: k-survivability: diversity and survival of expendable robots. IEEE Robot. Autom. Lett. **1**(2), 1164–1171 (2016)
11. Robin, C., Lacroix, S.: Taxonomy on multi-robot target detection and tracking. In: Workshop on Multi-Agent Coordination in Robotic Exploration (2014)
12. Sato, H., Royset, J.O.: Path optimization for the resource-constrained searcher. Naval Res. Logist. **57**(5), 422–440 (2010)
13. Schwager, M., Dames, P., Rus, D., Kumar, V.: A Multi-robot control policy for information gathering in the presence of unknown hazards, pp. 455–472. Springer, Cham (2017)
14. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
15. Vincent, P., Rubin, I.: A framework and analysis for cooperative search using UAV swarms. In: Proceedings of the 2004 ACM Symposium on Applied Computing, SAC 2004, pp. 79–86. ACM, New York (2004)

16. Yang, Y., Minai, A.A., Polycarpou, M.M.: Decentralized cooperative search by networked UAVs in an uncertain environment. In: American Control Conference, vol. 6, pp. 5558–5563. IEEE (2004)
17. Yang, Y., Polycarpou, M.M., Minai, A.A.: Multi-uav cooperative search using an opportunistic learning method. J. Dyn. Syst. Meas. Contr. **129**(5), 716–728 (2007)