Enhancing Search and Rescue Capabilities in Hazardous Communication-Denied Environments through Path-Based Sensors with Backtracking

Extended Abstract

Alexander Mendelsohn
University of Maryland
Maryland, United States
alexjmendelsohn@gmail.com

Don Sofge
Naval Research Laboratory
Maryland, United States
donald.sofge@nrl.navy.mil

Michael Otte
University of Maryland
Maryland, United States
ottemw@gmail.com

ABSTRACT

Prior work on path-based sensors has assumed that each agent's path is determined before the agent departs and cannot be changed mid-trip. We consider how an agent might adjust its path in response to new information that it gathers *en route*. Mid-trip path adjustment is non-trivial because it can increase the number of locations at which a missing agent may have been destroyed (from an external observer's point-of-view). We solve this issue by employing backtracking as a particular form of mid-trip path adjustment that avoids the issue of additional potential destruction locations.

KEYWORDS

Path-based sensors; Information Gathering; Path planning

ACM Reference Format:
Alexander Mendelsohn, Don Sofge, and Michael Otte. 2024. Enhancing Search and Rescue Capabilities in Hazardous Communication-Denied Environments through Path-Based Sensors with Backtracking: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Autonomous robots have been used to gather information in a variety of environments using all kinds of sensors. In some applications, we get a binary 'yes' or 'no' to the question of *whether* an event has occurred along the agent's path; however, when we get a 'yes,' we do not know *where* along the path the event occurred. This binary observation along a path is known as a "path-based sensor" [10].

We consider a a goal of finding the locations of both agent-destroying hazards and search and rescue targets in a communication denied environment. Searching for the hazards can be modeled as a path-based sensor because agents are unable to communicate the locations of their destruction to an outside observer. The targets can be located using a standard sensor (such as a camera). Because the environment is communication-denied, an agent's sensor readings about targets are lost if it is destroyed. This presents a dilemma. If the agent has accumulated valuable information but faces a high risk of future destruction, then it can be beneficial to deviate for a safer route. However, if we simply allow the agent to take the safest path back to base, then when an agent *is* destroyed the set of possible destruction locations expands to include every cell that could be part of some potential deviation back to the base. This would significantly dilute information gain and make the hazard belief update computationally intractable.

We propose a variant on the path-based sensor strategy that mitigates this issue by enabling agents to return to their starting point by reversing along their current path instead of completing the path to the goal (Fig. 1). Backtracking offers a balance between safety, informational value, and efficiency: the agent has just survived the sub-path it intends to use; it does not dilute the information gain; and it supports the use of computationally efficient update rules.

2 RELATED WORK

The concept of a path-based sensor is introduced in [10] and extended in [13]. The current paper differs from [10, 13] by allowing agents to adjust their paths mid-trip based on new information that is gathered *en route*. Our work differs from previous work [1–9, 11, 12, 14, 15] in that we consider a zero-communication scenario with multiple stationary hazards and targets where we cannot observe the locations where hazards destroy our agents.

3 ALGORITHM

The problem we are trying to solve has three parts:

a. Given an environment and beliefs about hazards and targets, find the path that maximizes the expected information gain.
b. Given a partially-executed plan, determine if backtracking will increase the expected information gain. (Alg. 1 line 3.)
c. Given a completed path or an agent disappearance along a planned path, update the belief map. (Alg. 1 lines 4-11.)

Algorithm 1 iterativeInfoGathering(X, Z)
Input: Prior target beliefs X and hazard beliefs Z
Output: Iterative sequence of paths, and updates to X and Z

```
1:  for r = 1, 2, ... do
2:    ζ ← calculatePath(X, Z)
3:    PathResult ← tryToTraversePath(ζ, X, Z)
4:    if PathResult = died then
5:      Z ← KilledOnPathUpdate(Z)
6:    else if PathResult = completed then
7:      X ← BayesianCellUpdates(X, Y_ζ)
8:      Z ← BayesianCellUpdates(Z, [0, ..., 0])
9:    else if PathResult = backtracked then
10:     X ← X′
11:     Z ← Z′
12: return X, Z
```

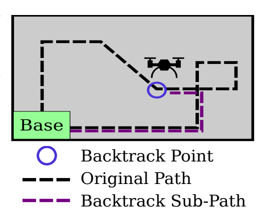


Figure 1: An agent backtracks and removes loops, improving safety.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

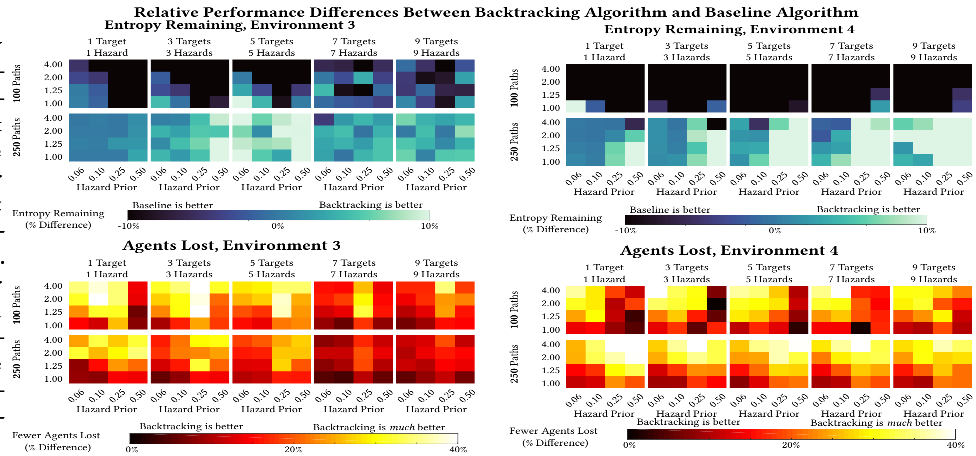


Figure 2: The difference in entropy remaining (top) and agents lost (bottom) when using backtracking versus the baseline algorithm. Dark and light squares indicate when the baseline or backtracking has better performance, respectively. Different (large) columns correspond to different numbers of targets and hazards. Large rows show performance after different numbers of paths. Within each heat-map: small columns correspond to hazard belief priors and the small rows correspond to backtracking bias . Backtracking tends to perform better and better over time.

Prior work focused on (a) and (c). **We explore (b), showing that allowing backtracking decreases the number of agents destroyed and often improves the rate of information gain.** We combine our method for (b) with existing solutions for (a) and (c).

The algorithm to find the locations of the hazards and targets is an iterative approach that plans a path for each agent based on the current hazard and target belief states, sends the agent along the path, and updates the belief states based on whether the agent survived as well as its target sensor observations if it did (see Alg. 1).

At each cell along its path, an agent checks for a target and then decides whether to backtrack or continue on its path based on which has a higher expected information gain. Let c represent continuing forward, b represent backtracking, d represent agent destruction, and I represent information gained. Furthermore, let:

(1) $\mathbf{P}(d|c)$: probability of destruction if agent continues.
(2) $\mathbf{E}(I|d)$: expected info if agent is destroyed.
(3) $\mathbf{E}(I|\neg d \wedge c)$: expected info if agent survives continuing.
(4) $\mathbf{P}(d|b)$: probability of destruction if agent backtracks.
(5) $\mathbf{E}(I|\neg d \wedge b)$: expected info if agent survives backtrack.

This gives $\mathbf{E}(I|c) = \mathbf{P}(d|c)\mathbf{E}(I|d) + (1 - \mathbf{P}(d|c))\mathbf{E}(I|\neg d \wedge c)$ and $\mathbf{E}(I|b) = \mathbf{P}(d|b)\mathbf{E}(I|d) + (1 - \mathbf{P}(d|b))\mathbf{E}(I|\neg d \wedge b)$. The agent backtracks if $\mathbf{E}(I|b) > (bias)\mathbf{E}(I|c)$.

$\mathbf{E}(I|d)$ is the same whether or not the agent backtracks, and has already been calculated during the path planning. To compute $\mathbf{P}(d|c)$, we start with the probability of surviving the entire path, which is the product of the probabilities of surviving each cell. The probability of surviving a given cell is $((1 - Z[c]) + Z[c](1 - lethality))(1 - malfunction)$ where $Z[c]$ is the hazard prior for cell c. Then after each cell the agent survives, we update this probability based on the fact that the remaining path is one cell shorter.

Given our assumption of cell independence, the expected information gained during the backtrack is the sum of the expected information gained in each cell. $\mathbf{E}(I|\neg d \wedge b)$ is found by adding the expected information that will be gained during the backtrack to the information already gained along the forward path.

When an agent survives a path, it necessarily survived every cell along that path. So, in the case of agent survival, hazard belief updates are achieved using a standard bayesian update independently for each cell visit. Similarly, target beliefs are updated independently for each target sensor reading obtained along the path.

When an agent is destroyed on a path, however, we do not know where it was destroyed. We can use our priors to calculate the probability that the agent was destroyed in each cell. Then for each possible location, we use standard bayesian updates to calculate the posterior probabilities that would result if we knew the agent had been destroyed in that location. Finally, we take the weighted average of those possibilities to compute our combined posterior.

4 EXPERIMENTS AND CONCLUSIONS

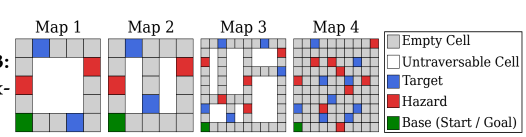
We run simulations to test how the algorithm performs in the four environments shown in Fig. 3. We compare allowing backtracking to the baseline (non-backtracking) algorithm, and evaluate how performance is affected by the number of targets and hazards, the hazard lethality rate, the bias in favor of (or against) backtracking, and the prior probability of hazards or targets assumed in each cell.

Our experiments show that backtracking resulted in fewer agent losses than the baseline algorithm in nearly all trials — 20% fewer agents lost in most cases, with up to 40% fewer agents lost in some cases (Fig. 2). We observe that backtracking converges to low entropy faster than the baseline in some cases. In the larger environments, the backtracking algorithm starts off gathering information more slowly than the baseline method, but then catches up and outperforms the baseline method as more agents are deployed. Overall, backtracking algorithm appears to be an improvement over the baseline algorithm when a large number of paths (approximately 100-200) can be run, or when losing fewer agents is a priority.

ACKNOWLEDGMENTS

This work is supported by the Maryland Robotics Center and ONR grants N0001420WX01827 and N00014-20-1-2712. The views, positions, and conclusions contained in this document are solely those of the authors and do not explicitly represent those of ONR.

Figure 3: Maps for experiments.



REFERENCES

[1] Akash Arora, P Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. 2019. Multi-modal active perception for information gathering in science missions. *Autonomous Robots* 43, 7 (2019), 1827–1853.

[2] J Berger and J Happe. 2010. Co-evolutionary search path planning under constrained information-sharing for a cooperative unmanned aerial vehicle team. In *Evolutionary Computation, IEEE Congress on*. 1–8.

[3] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. 2011. Search and pursuit-evasion in mobile robotics. *Autonomous Robots* 31, 4 (2011), 299–316.

[4] Philip Dames and Vijay Kumar. 2015. Autonomous Localization of an Unknown Number of Targets Without Data Association Using Teams of Mobile Sensors. *Automation Science and Engineering, IEEE Transactions on* 12, 3 (2015), 850–864.

[5] P. Dames, M. Schwager, V. Kumar, and D. Rus. 2012. A decentralized control policy for adaptive information gathering in hazardous environments. In *IEEE Conference on Decision and Control*. 2807–2813. https://doi.org/10.1109/CDC.2012.6426239

[6] Geoffrey A Hollinger, Srinivas Yerramalli, Sanjiv Singh, Urbashi Mitra, and Gaurav S Sukhatme. 2015. Distributed data fusion for multirobot search. *Transactions on Robotics* 31, 1 (2015), 55–66.

[7] Brian J Julian, Michael Angermann, Mac Schwager, and Daniela Rus. 2012. Distributed robotic sensor networks: An information-theoretic approach. *The International Journal of Robotics Research* 31, 10 (2012), 1134–1154.

[8] Asif Khan, Evsen Yanmaz, and Bernhard Rinner. 2015. Information exchange and decision making in micro aerial vehicle networks for cooperative search. *Control of Network Systems, IEEE Transactions on* 2, 4 (2015), 335–347.

[9] Sangwoo Moon and Eric W. Frew. 2019. A Communication-Aware Mutual Information Measure for Distributed Autonomous Robotic Information Gathering. *IEEE Robotics and Automation Letters* 4, 4 (2019), 3137–3144. https://doi.org/10.1109/LRA.2019.2924850

[10] Michael Otte and Don Sofge. 2021. Path-Based Sensors: Paths as Sensors, Bayesian Updates, and Shannon Information Gathering. *IEEE Transactions on Automation Science and Engineering* 18, 3 (2021), 946–967. https://doi.org/10.1109/TASE.2021.3067628

[11] Cyril Robin and Simon Lacroix. 2014. Taxonomy on Multi-robot Target Detection and Tracking. In *Workshop on Multi-Agent Coordination in Robotic Exploration*.

[12] Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar. 2017. *A Multi-robot Control Policy for Information Gathering in the Presence of Unknown Hazards*. Springer International Publishing, Cham, 455–472.

[13] Alkesh Kumar Srivastava, George P Kontoudis, Donald Sofge, , and Michael Otte. 2022. Distributed Multi-Robot Information Gathering using Path-Based Sensors in Entropy-Weighted Voronoi Regions. In *International Symposium on Distributed Autonmous Robotic Systems (DARS)*. Montbéliard, France.

[14] Maciej Stachura and Eric Frew. 2017. Communication-Aware Information-Gathering Experiments with an Unmanned Aircraft System. *Journal of Field Robotics* 34, 4 (2017), 736–756.

[15] John Tisdale, ZuWhan Kim, and J Karl Hedrick. 2009. Autonomous UAV path planning and estimation. *Robotics & Automation Magazine, IEEE* 16, 2 (2009), 35–42.