

# Soil Sample Search in Partially Observable Environments

Han Yang<sup>1</sup>, Andrew Dudash<sup>2</sup>

**Abstract**—To work in unknown outdoor environments, autonomous sampling machines need the ability to target samples despite limited visibility and robotic arm reach distance. We design a heuristic guided search method to speed up the search process and more efficiently localize the approximate center of soil regions. Through simulation experiments, we assess the effectiveness of the proposed algorithm and discover superior performance in terms of speed, distance traveled, and success rate compared to naive baselines.

## I. INTRODUCTION

In this paper, we address the problem of autonomous sample collection in outdoor, unknown environments. While collecting soil or similar organic material, there are no guarantees that samples will be reachable, visible, or even present. For this reason, a robot needs an effective search task to locate and move sufficiently close to the samples prior to beginning collection. We formulate this problem as a target search problem for one of multiple irregularly shaped, i.e. amorphous, target regions in a heterogeneous terrain, similar to how stretches of open soil appear in nature.

Specifically, we employ a heuristic guided method to locate the centers of soil clusters containing minimum non-pickable obstacles, like rocks or grass. This search method assumes the camera provides a bird's eye field of view, but can be applied to a variety of sample collection applications: an unmanned aerial vehicle that identifies and reports large regions containing samples or the vision system of a unmanned ground vehicle moving to pick an obstacle. While this method may not be optimal, it presents a simple to implement algorithm that requires no prior knowledge about target distribution, and has experimentally been found effective and robust against visibility impairments due to factors such as fog and dust. The contributions of this paper are as follows:

- We develop a heuristic that allows autonomous robots to find the center of an amorphous target in a 2D environment, with applications to soil and organic material collection in heterogeneous outdoor terrains.
- We experimentally measure the effectiveness of our heuristic method compared to a variation on it and two offline-generated search patterns. Our experiments included over 5000 trials across different soil abundance levels and visibility conditions.

<sup>1</sup> Noblis Autonomous Systems Research Center Reston, Virginia  
Han.Yang@noblis.org

<sup>2</sup> Noblis Autonomous Systems Research Center Reston, Virginia  
Andrew.Dudash@noblis.org

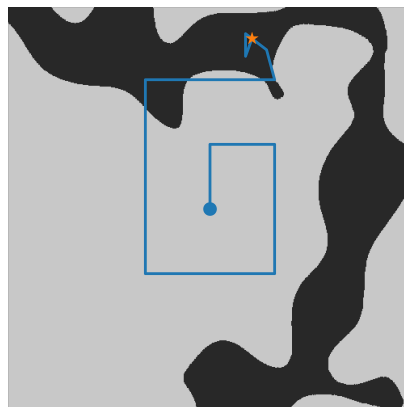


Fig. 1: In this example, a robot—perhaps a camera mounted end effector of a robotic arm—uses a heuristic method to search for the center of a soil region in a sample distribution. The circle is the start position, and the star indicates the center which the agent aims to reach.

## II. RELATED WORK

The rapid development of autonomous robotics has drastically improved the efficiency and success rate for target search and tracking problems due to improved maneuverability, versatility, and ability to make decisions with limited information. Such systems have been applied to a multitude of fields including home services [1], marine maintenance [2], field exploration [3], and wilderness search and rescue (SAR) [4], [5].

Searching for a single, static object is the simplest form of the target search problem. For this case, a popular class of solutions discretizes the search domain through methods such as common grid-based cellular decomposition [6] and graph-based representation. [7]. Compared to continuous state space methods, these approaches simplify optimal solution computation. For example, with discretization, it is easier to formulate solutions to dynamic target problems—like partially observable Markov decision processes (POMDP) [1]—because the motion model is also simplified. Moreover, a paper in frontier exploration [8], [9] decompose the search region into square grids to simplify frontier and entropy map representation. Nevertheless, in our problem with amorphous target regions, the discretization approach may result in partially filled grids that are difficult to classify, as well as multiple fully filled grids that cannot be tackled with single target search approaches.

Another popular class of solutions represents the problem using a continuous state space as studied by [10], [11]. Both papers use deep Q-network (DQN) designs compatible with continuous state spaces for UAV SAR and surveillance tasks.

In problems with non-trivial low-level dynamics, adopting a continuous solution helps to ensure the dynamic feasibility of planned paths [12]. Still, these approaches focus on a single point-based target whose state can easily be represented by a small number of coordinates, which can be challenging to extend into problems involving amorphous targets.

Many papers have generalized the search problem to locating multiple targets. When the number of targets is known in advance, RL and MDP based approaches can efficiently locate all targets that are either static or dynamic [2], [4] with little online computation time. Luo et al. and Vanegas et al. studied the multi-target search problem with static obstacles in the search region to better extend the solution to outdoor search missions [13], [14]. While such approaches are extremely time efficient, they require priori knowledge on the target’s distribution and do not scale well to problems without a well-defined number of targets.

Coverage path planning (CPP) is a natural solution to locate all the targets in the search region without information regarding target quantities. A prominent class of such approach involves offline, deterministic path planning. An example is the lawnmower search pattern, which has been officially standardized for marine-time search and rescue [15]. While such methods can guarantee success in most cases, they are often less efficient and cannot be naively employed without priori information about the search space. In this regard, Ianenko et. al. [16] and Champagnie et. al. [17] developed proximal policy optimization and multi-agent entropy maximization based CPP approaches that more efficiently cover an unknown environment with dynamic and static obstacles.

In our problem, the objective is to locate a singular target region, and full coverage approaches that search for every possible target in the domain may lead to unnecessary effort. In addition, most approaches assume point-based target models, which may fail to capture the amorphous nature of our search targets. Although treating the target areas as clusters of points may work in conjunction with search techniques developed for an unknown number of targets, for example, that detailed in [18], it does not exploit clustering information to improve search efficiency. Therefore, we seek a solution that accounts for the amorphous and clustered nature of soil beds.

Lastly, unlike many search path planning methods that assumes a fixed field of view [2], [4], [16], [17], we leverage extra information that can be gathered by simply increasing the camera’s height and thus the view size, similar to methods studied by Gupta et. al., Bhagat et. al., Goodrich et. al [5], [6], [11]. In particular, Gupta and Goodrich each assume that the resolution of the camera’s view scales linearly with height. However, in our implementation, we consider other factors such as fog and dust which are prevalent in outdoor environments and may further impair visibility.

### III. APPROACH

#### Overview

In our problem formulation, we make simplifying assumptions that state transitions and position feedback are perfect and that the search region is open and free of obstacles. The state space  $\mathbb{S}$  of the robot is a 3 tuple containing the possible positions of the agent  $p = (x_A, y_A, z_A)$ . Given limitations on the camera’s resolution and physical bounds of a robot’s workspace, we constrain the robot’s height to be between  $z_A \in [h_{\min}, h_{\max}]$  and the location  $(x_A, y_A)$  to be within some bounds  $x_A \in [x_{\min}, x_{\max}]$ ,  $y_A \in [y_{\min}, y_{\max}]$ . In addition, the observation space  $\mathbb{O}$  includes the target’s location and the robot’s view. At each time step, the robot receives a perfect measurement of its position  $p$  and fixed size, noisy image  $\tilde{I}$ . Similar to sampling methods implemented by [19], we make the image a grayscale segmentation plot that represents whether a pixel contains soil samples or not. For simplicity, we convert the segmented image into a binary valued matrix,  $I_b$ , given some threshold  $t$ , represented by Equation 1.

$$I_{b,(i,j)} = \begin{cases} 0, & \tilde{I}_{(i,j)} < t, \\ 1, & \tilde{I}_{(i,j)} \geq t. \end{cases} \quad (1)$$

To leverage the variable zoom size that can be achieved by our system, we include an action in the  $z$  direction such that the agent can scan a larger or smaller area when needed. Therefore, we design the action space  $\mathbb{A}$  to be a 3 tuple  $a = (\Delta x, \Delta y, \Delta z)$ , corresponding to changes in the agent’s position.

#### Heuristic Policy

Our search strategy follows a heuristic on the soil cluster’s center given the current observation, as shown in Figure 2. At each time step, given a binary valued image  $I_b$ , we first compute the point and direction towards which the agent would move. This point is defined as the center of soil samples viewed by the robot and is computed by tracing the contours of observed soil clusters, then computing the centroid of the largest contour using Equation 2,

$$A\bar{x} = \sum_{x=0}^n \sum_{y=0}^n x I_{c,(y,x)} \quad (2)$$

$$A\bar{y} = \sum_{y=0}^n \sum_{x=0}^n y I_{c,(y,x)}, \quad (3)$$

where  $A$  is the area of the largest contour,  $I_c$  is a copy of  $I_b$  with only the largest contour, and  $(\bar{x}, \bar{y})$  is the centroid in image coordinates. Afterwards, we take a greedy step by assuming that the center of the soil region is in the direction of the computed centroid. Therefore, the locally optimal solution would be to move in the relative direction between the current location, which is the center of the image frame, and the centroid, given by Equation 4.

$$\Delta x = \gamma(\bar{x} - x_A) \quad (4)$$

$$\Delta y = \gamma(\bar{y} - y_A), \quad (5)$$

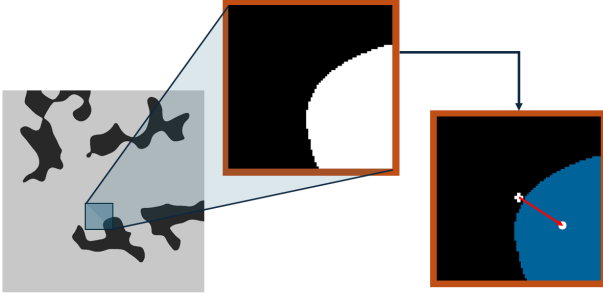


Fig. 2: In this illustration, the contour of the spotted soil samples are shown as the blue region in the rightmost image. The centroid, indicated by the white dot, is determined from the contour and used to define the next waypoint and action, shown by the red arrow.

where  $\gamma$  is a scalar zoom size dependant on  $z_A$  to convert the step into world coordinates. In addition to traversing in the 2D plane, once soil samples are spotted, the robot would lower its height to increase the observation accuracy, similar to the logic of a typical human pilot in SAR operations.

In situations where no soil samples are spotted, the agent searches for targets through exploration. The notion of exploration here applies not only to exploring un-visited regions in the 2D plane, but also descending in height to further verify whether the current location contains soil samples. In the first case, labeled as  $A_{\text{explore}}$ , we program the agent to follow a deterministic baseline path, such as the expanding square pattern described in [15], while ascending for a larger viewing area. In the latter case  $A_{\text{dive}}$ , the robot would descend vertically without moving along the x-y plane.

To balance between trusting the observation or not, the agent chooses either one of the two actions in a probabilistic manner. The decision probability depends on the expected accuracy of the received observation, given by Equation 6.

$$P(A_{\text{explore}}) = f(\hat{z}_A, \theta) \quad (6)$$

$$P(A_{\text{dive}}) = 1 - f(\hat{z}_A, \theta), \quad (7)$$

where  $\hat{z}_A$  is the normalized height, and  $\theta$  represent visibility parameters. While the true relation between observation accuracy, visibility, and height is unknown, for simplicity, we define the model to be parameterized by a scalar visibility factor and set the decision probability given by Equation 8.

$$f(\hat{z}_A, \theta) = \frac{e^{\hat{z}_A/\theta}}{e^{\hat{z}_A/\theta} + 1}. \quad (8)$$

#### IV. EXPERIMENTS

We validate our search strategy in a simulated 2D environment with a randomized distribution of soil samples and a static initial position for our agent, similar to the setup in [2]. We measure and compare the step count, distance traveled, and success rate of four different search methods under two different visibility scenarios.

---

#### Algorithm 1 Heuristic Search Method

---

```

while not in soil region do
  receive observation  $o = \{p, I_b\}$ 
  if no target spotted then
    compute  $\hat{z}_A$  and  $f(\hat{z}_A, \theta)$ 
    descend with probability  $f(\hat{z}_A, \theta)$ 
    follow pattern with probability  $1 - f(\hat{z}_A, \theta)$ 
  else
    find contours from  $I_b$ 
    find centroid  $(\bar{x}, \bar{y})$  of the largest contour
    move towards centroid and ascend
  end if
  action =  $(\Delta x, \Delta y, \Delta z)$ 
end while

```

---

#### Simulation

We developed a simulation environment using the Gymnasium package in Python [20]. The environment is saved as a 1000x1000 image, with each pixel occupying a binary value that indicates whether the pixel contains a soil sample.

We utilize the OpenCV library [21] to generate amorphous and random target areas. Specifically, we start by generating a white noise image that is then transformed into rough contours with the `GaussianBlur` method. The outline of the contours is then highlighted by re-scaling the intensity and applying a binary threshold. Lastly, holes in the target areas are closed through the `morphologyEx` method. To avoid the trivial case, we do not spawn the robot within a target region. An illustration of the generated environment is shown in Figure 3.

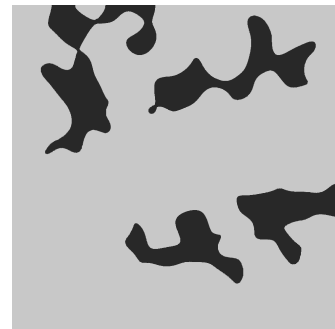


Fig. 3: In this example, a simulation environment with an amorphous soil distribution is shown. The dark regions represent soil, that can be sampled, and the lighter regions represent all other materials.

For each step of the experiment, the search agent outputs an action to the environment, after which the environment outputs observation information. The observation contains the current location, in pixels, of the robot, the robot's view as a 25x25 image, the distance traveled, and the step count, which will later be used as metrics for search performance.

Note that although the size of the robot's view size changes with respect to height, much like a real camera, the search policy expects images with fixed dimensions,

thus all observations are resized before being passed to the agent. In addition, to simulate sensor noises and visibility impairment, we add artificial noise to the robot’s view using OpenCV’s `addWeighted` method. Given that impaired visibility would generally result in false negative detection (failure to detect soil samples), the noise added is set to be predominantly 0, as shown in Figure 4.

Similar to Equation 8, the robot’s visibility is parameterized by a single parameter that dictates the shape of a sigmoid blending function, given by Equation 9.

$$\alpha = \frac{e^{\hat{z}_A/\theta}}{e^{\hat{z}_A/\theta} + 1} + \nu, \quad (9)$$

where  $\alpha$  is the weight of the noisy image,  $\nu$  is a normal random scalar with mean of 0 and variance of 0.15.

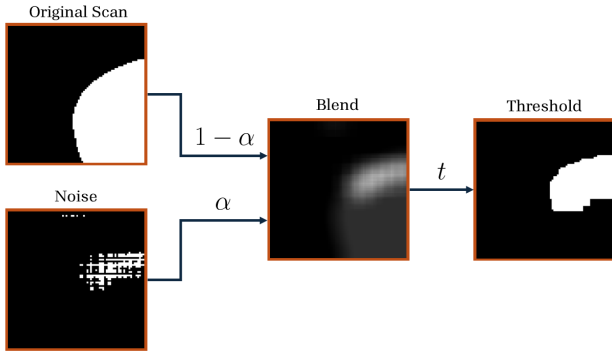


Fig. 4: To artificially impair the robot’s visibility, we add noise, weighted by  $\alpha$ , to the ground truth view. The blended image is then passed through a threshold to obtain a binary valued image. In this example, parts of the original soil region are hidden from the agent by noise.

Each trial starts with the robot spawning at the center of the map with a height of 50% of  $h_{\max}$ , and terminates under one of three conditions:

- *Target Found*: The robot identifies the central region of one of the clusters with high certainty, quantified by a scan with above 95% soil and the robot at height below 10%.
- *Truncated*: The step number exceeds 300.
- *Out of Bounds*: The robot attempts to move to a point outside of the 1000x1000 environment.

We implemented four search methods, two baselines and two versions of our heuristic method. We use the baselines to benchmark the performance of our proposed algorithm. Each method is detailed below:

- *Expanding Square*: The expanding square is a deterministic search pattern suggested by [15] and is commonly used in search and rescue operations.
- *Lissajous Curve*: The Lissajous curve is a deterministic search pattern proposed by [22], who found the method to be faster than traditional search patterns while still being comprehensive. In our implementation, we take linearly spaced intervals of the parameter  $t$  to generate points at which the robot will make a scan.

- *Heuristic + Expanding Square*: This approach combines the proposed method with an expanding square pattern.
- *Heuristic + Lissajous Curve*: This approach combines the proposed method with a Lissajous pattern.

### Search Performance Experiments

We evaluated the performance of our algorithm under two scenarios: high environment visibility ( $\theta = 0.7$ ), and low environmental visibility ( $\theta = 0.2$ ). The detection accuracy for each test case is plotted in Figure 5.

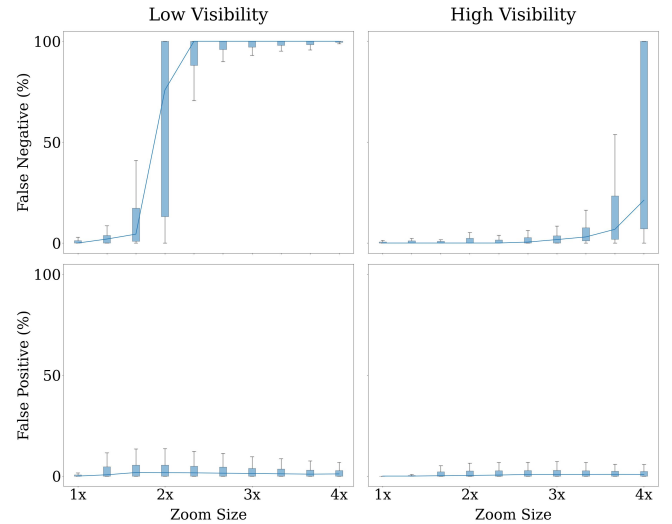


Fig. 5: The  $y$  values correspond to the percentage of pixels that are misclassified. In a low visibility environment, nearly all of the soil samples would be missed at a zoom size of above 2x, corresponding to around 40% between  $h_{\max}$  and  $h_{\min}$ , and around 1-2% of the remaining region would be detected as soil. On the other hand, in a high visibility environment, only a small portion of the soil samples will be missed in most cases, while the false positive rate remains at a similar level.

We run 5000 simulation trials for each test scenario and method, and we record the following metrics:

- *Number of Steps*: The number of steps in a trial before a robot finds a target. This roughly scales with the time taken, as each step comes with the time overhead of scanning, image processing, and decision making.
- *Distance Traveled*: The distance traveled by the agent in successful trials before finding a target sample.
- *Status*: Whether the robot finds a target. If it fails, the status includes the failure mode.

## V. RESULTS

From the experiments, our proposed method successfully demonstrated superior performance for all three metrics, and was robust to different visibility levels.

### High Visibility Experiment

In the high visibility case, we discovered that the heuristic guided search methods outperform the deterministic baseline pattern with respect to all three metrics, especially

in environments with a sparse amount of soil, as shown in Figure 6. This outcome matched our expectations as the heuristic methods ideally would converge to the center of soil regions as soon as a small fraction of the soil is detected. The heuristic methods with the two baselines performs similarly in terms of number of steps across the soil abundance spectrum, but the method with a Lissajous baseline took only around 80% of traveled distance in sparse environments. In terms of success rate, the naive expanding square pattern experienced a significant number of failures in low soil abundance environments, which can be attributed to the large chosen step size that allows little to no overlap between two successive scans and causes smaller target areas that lie between to be omitted. On the other hand, the Lissajous-based method, both the naive and heuristic one, were able to achieve a success rate of above 98% regardless of soil abundance. We identified two main failure modes of our proposed algorithm with a Lissajous baseline, indefinite looping and moving out of bounds:

- In an indefinite looping scenario, the agent would spot a small soil region that is smaller than the minimum defined step size. As a result, the search algorithm will "overshoot" a step and the target would be in the direction which the agent came from, leading to an indefinite movement back and forth about the target region and a *Truncated* final status.
- In an out of bounds case, the robot would fail to detect any soil regions along its search pattern, either due to false negative detection, or soil samples lying in between the gaps of two successive scans, leading to an *Out of Bounds* final status.

Figure 8 illustrates the proportion of each failure modes in a soil sparse environment.

#### Low Visibility Experiment

In the low visibility experiments, we discovered that the proposed heuristic still improved the performances of the baseline patterns despite the changed visibility, shown in Figure 7. The naive baselines performed similarly in this experiment as they did in the high visibility experiment. This matched our expectations as they were programmed to operate at the lowest height where noise has minimal effect. On the other hand, we found that visibility impairment only mildly degraded the performances of the heuristic methods. There was a slight success rate drop of 2.6% for the Lissajous-based heuristic method, while the step number and total distance traveled only increased by a small margin, as shown in Figure 9. This minor variation between the two experiments showcases the robustness of our proposed method against different visibility conditions with little hyper-parameter tuning required.

### CONCLUSION

We studied the problem of amorphous target search, which has application such as soil collection. We proposed a heuristic that drives the robot towards the center of a soil region

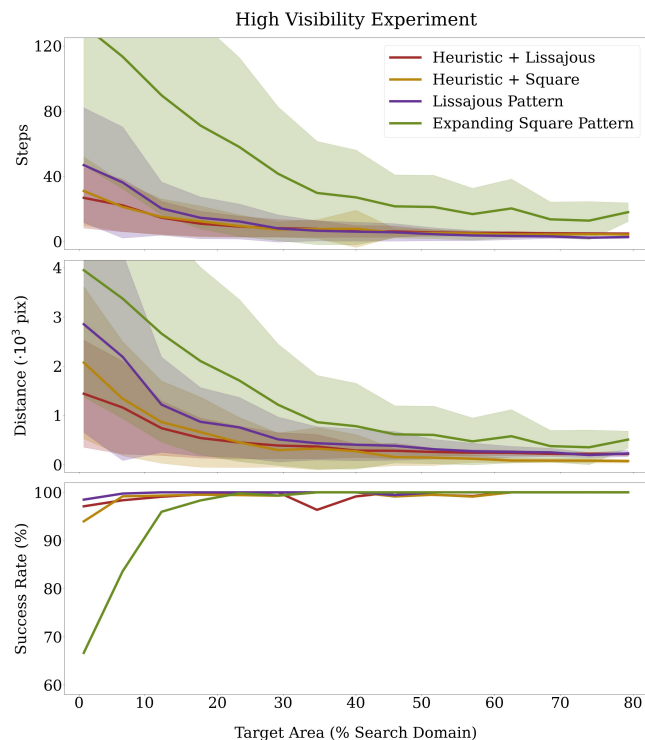


Fig. 6: Among the four tested method, the Lissajous-based heuristic method outperforms all other methods in terms of step count and distance traveled while maintaining a near 100% success rate across different soil abundance levels.

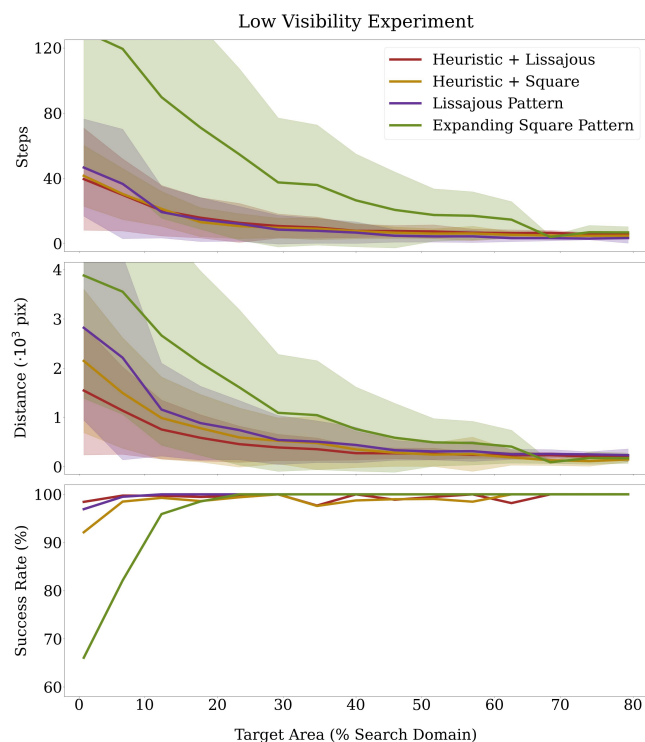


Fig. 7: Similar to the high visibility case, the Lissajous-based heuristic method demonstrates superior performance compared to the other three methods.

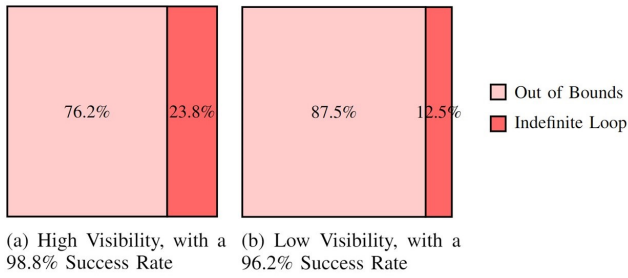


Fig. 8: The proportions of each failure mode under different visibility conditions in environments with less than 15% soil are shown. Out of bound failures are more common in low visibility conditions, which is caused by the higher probability of missed detection.



Fig. 9: The performances of the Lissajous-based heuristic method under two different visibility levels. In environments with low soil level, the agent on average took only 10 more steps to locate targets with low visibility than with high visibility.

using a camera with a limited field of view. Through simulation experiments, we verified that the proposed algorithm outperforms patterns used in search and rescue missions. The Lissajous-based heuristic search resulted in the least amount of steps and distance traveled across different soil abundance and visibility levels.

This research can further be extended by incorporating frontier based exploration methods into our heuristic, and training the system with reinforcement learning to further improve resilience against visibility uncertainties. Lastly, an important future research direction is to further verify the effectiveness of this method through hardware experiments and field tests.

## REFERENCES

- [1] D. Hsu, W. S. Lee, and N. Rong, "A point-based pomdp planner for target tracking," in *2008 IEEE International Conference on Robotics and Automation*, (Pasadena, CA, USA), p. 2644–2650, IEEE, May 2008.
- [2] M. Meghjani, S. Manjanna, and G. Dudek, "Multi-target search strategies," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, (Lausanne, Switzerland), p. 328–333, IEEE, Oct. 2016.
- [3] Y. Zhao, L. Yan, H. Xie, J. Dai, and P. Wei, "Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor," *IEEE Transactions on Industrial Electronics*, vol. 71, p. 4933–4943, May 2024.
- [4] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, and X. Liang, "Uav autonomous target search based on deep reinforcement learning in complex disaster scene," *IEEE Access*, vol. 7, p. 117227–117245, 2019.
- [5] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.
- [6] A. Gupta, D. Bessonov, and P. Li, "A decision-theoretic approach to detection-based target search with a uav," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Vancouver, BC), p. 5304–5309, IEEE, Sept. 2017.
- [7] Z. Botao, W. Jun, L. Qiang, and S. A. Chepinskiy, "A expected-time optimal path planning method for robot target search in uncertain environments," in *2018 37th Chinese Control Conference (CCC)*, (Wuhan), p. 5554–5559, IEEE, July 2018.
- [8] M. Kim, R. Gupta, and L. Sentis, "Information-theoretic based target search with multiple agents," July 2021. arXiv:2107.12715 [cs].
- [9] B. Yamauchi, "A frontier-based approach for autonomous exploration," (Monterey, CA, USA), p. 146–151, 1997.
- [10] X. L. Wei, X. L. Huang, T. Lu, and G. G. Song, "An improved method based on deep reinforcement learning for target searching," in *2019 4th International Conference on Robotics and Automation Engineering (ICRAE)*, (Singapore, Singapore), p. 130–134, IEEE, Nov. 2019.
- [11] S. Bhagat and P. Sujit, "Uav target tracking in urban environments using deep reinforcement learning," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, (Athens, Greece), p. 694–701, IEEE, Sept. 2020.
- [12] B. Li and Y. Wu, "Path planning for uav ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, p. 29064–29074, 2020.
- [13] Y. Luo, Z. Zhuang, N. Pan, C. Feng, S. Shen, F. Gao, H. Cheng, and B. Zhou, "Star-searcher: A complete and efficient aerial system for autonomous target search in complex unknown environments," *IEEE Robotics and Automation Letters*, vol. 9, p. 4329–4336, May 2024. arXiv:2402.16348 [cs].
- [14] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, "Uav based target finding and tracking in gps-denied and cluttered environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Daejeon, South Korea), p. 2307–2313, IEEE, Oct. 2016.
- [15] I. M. O. C. A. Organization, *IAMSAR Manual, Volume I – Organization and Management*. International Maritime Organization, 2022 edition ed., 2022.
- [16] A. Ianenko, A. Artamonov, G. Sarapulov, A. Safaraleev, S. Bogomolov, and D.-k. Noh, "Coverage path planning with proximal policy optimization in a grid-based environment," in *2020 59th IEEE Conference on Decision and Control (CDC)*, (Jeju, Korea (South)), p. 4099–4104, IEEE, Dec. 2020.
- [17] K. Champagne, F. Arvin, and J. Hu, "Decentralized multi-agent coverage path planning with greedy entropy maximization," in *2024 IEEE International Conference on Industrial Technology (ICIT)*, (Bristol, United Kingdom), p. 1–6, IEEE, Mar. 2024.
- [18] B. Yousuf, Z. Lendek, and L. Buşoniu, "Exploration-based search for an unknown number of targets using a uav," *IFAC-PapersOnLine*, vol. 55, no. 15, p. 93–98, 2022.
- [19] A. Dudash, B. Andrades, R. Rubel, M. Goli, N. Clark, and W. Ewald, "Autonomous soil collection in environments with heterogeneous terrain."
- [20] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023.
- [21] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [22] J. J. Steckenrider, M. Miller, R. Blankenship, V. Trujillo, and J. Bluman, "Lissajous curves as aerial search patterns," *Scientific Reports*, vol. 14, p. 11144, May 2024.