

# Multi-Agent Active Search using Detection and Location Uncertainty

Arundhati Banerjee<sup>1</sup>, Ramina Ghods<sup>1</sup>, Jeff Schneider<sup>1</sup>

**Abstract**—Active search, in applications like environment monitoring or disaster response missions, involves autonomous agents detecting targets in a search space using decision making algorithms that adapt to the history of their observations. Active search algorithms must contend with two types of uncertainty: detection uncertainty and location uncertainty. The more common approach in robotics is to focus on location uncertainty and remove detection uncertainty by thresholding the detection probability to zero or one. In contrast, it is common in the sparse signal processing literature to assume the target location is accurate and instead focus on the uncertainty of its detection. In this work, we first propose an inference method to jointly handle both target detection and location uncertainty. We then build a decision making algorithm on this inference method that uses Thompson sampling to enable decentralized multi-agent active search. We perform simulation experiments to show that our algorithms outperform competing baselines that only account for either target detection or location uncertainty. We finally demonstrate the real world transferability of our algorithms using a realistic simulation environment we created on the Unreal Engine 4 platform with an AirSim plugin.

## I. INTRODUCTION

Active search [1], [2], in applications like wildlife patrolling and environment monitoring, involves autonomous robots (agents) discovering objects of interest (OOI) in an unknown environment by adaptively making sequential data collection decisions [3], [4], [5]. In such real world scenarios, active search algorithms are faced with two types of uncertainty: detection uncertainty and location uncertainty of observed OOIs. Due to noise in the sensing setup, an agent may mistake something that is not a target to be an OOI or vice versa, thereby giving rise to detection uncertainty. Further, systematic errors in the sensors may cause an agent to perceive OOIs to be located nearer to or farther from itself than they actually are, leading to uncertainty about the target's true location. Prior research has typically addressed only one of these while assuming no uncertainty in the other. In robotics, localization and mapping or tracking algorithms consider only target location uncertainty while abstracting away false positive or negative target detections [6], [7]. On the other hand, adaptive compressive sensing [8], [9], Bayesian optimization [10], [11] and bandit style algorithms [12], [13] only consider uncertainty in target detection while assuming perfect localization. Unfortunately, none of these algorithms present a unified framework accounting for both target detection and location uncertainty in active search.

Besides facing observation uncertainty, multi-agent active search introduces an additional challenge. Centralized plan-

ning with multiple agents is common but often impractical due to communication constraints [3], [14]. Further, a system dependant on a central coordinator expecting synchronicity from all agents is also susceptible to communication or agent failure. In our problem formulation, the agents share information with their peers when possible, but do not depend on any particular communication arriving. We assume that each agent independently plans and executes its sensing actions using whatever information it already has or receives. **Contributions** In this paper, we introduce a realistic sensing model that accounts for uncertainty in both target detection and location. Using these noisy observations, we propose an online inference algorithm called UnIK (Uncertainty-aware Inference using Kalman filter) for estimating the presence and position of targets in an unknown environment. Using UnIK, we formulate a Thompson sampling (TS) based multi-agent active search algorithm called TS-UnIK to enable decentralized decision making with asynchronous communication. We provide simulation results showing that under an identical set of sensing measurements, UnIK significantly outperforms other inference methods that only consider either target detection or location uncertainty. We provide simulation results demonstrating the efficiency of the action selection strategy in TS-UnIK. Finally, we demonstrate TS-UnIK in a realistic simulation environment created on the Unreal (UE4) platform with Airsim plugin. We implement OOI detection and location uncertainty modeling and show the real-world transferability of TS-UnIK using a ground robot in this setup.

## II. PROBLEM DEFINITION

Consider multiple ground robots searching a space to locate some OOIs (Fig. 1a). Each robot moves around and observes certain regions of the search space by taking pictures to detect OOIs and measuring their distances from its current location. We assume the robots can localize themselves accurately. The colored cells in Fig. 1a illustrate each robot's sensing action using a camera with a 90° field of view (FOV). Each robot independently decides its next sensing action given its current belief about OOIs in the search space.

Once a robot senses a region, it obtains information about both the presence (detection) and corresponding depth (location) of objects in its FOV. For example, RGB-D sensors coupled with an object detector like YOLOv3 [15] can be used to extract such information. The object detector identifies OOIs with a confidence score that varies with distance from the camera. Objects farther away from the camera usually have a lower probability of being correctly identified. This gives rise to target detection uncertainty in the robot's observation. Further, the depth sensor is prone

<sup>1</sup>A. Banerjee, R. Ghods and J. Schneider are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. {arundhat, rghods, schneide}@cs.cmu.edu

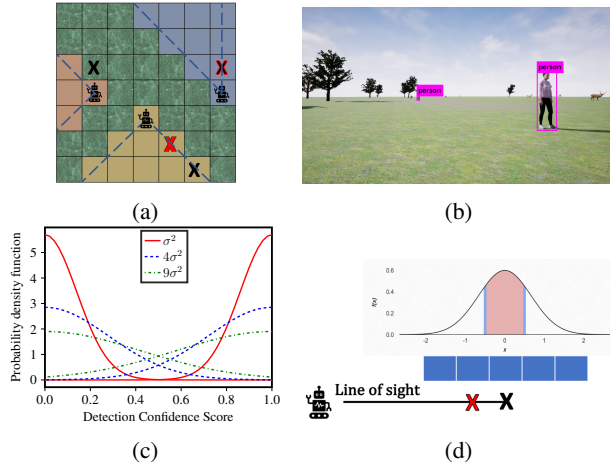


Fig. 1: (a) Multiple agents sense different parts of the environment looking for OOIs. True OOIs are crossed in black. Targets detected by the agent in its field of view are crossed in red. (b) OOI detections using YOLOv3 [15] in the robot’s FOV in our UE4 simulator (Section V-C). (c) An object detector becomes less confident about positive (1) as well as negative (0) labels as its distance to the object increases (Eq. (1)). (d) Due to location uncertainty, the observed depth of the true OOI (“X” in black) is shifted towards the agent, to the grid cell marked “X” in red.

to error in the measured distance to the OOI. This leads to target location uncertainty in the robot’s observation. Our objective is to account for both the detection and location uncertainty in our observation model and utilize it to make improved active sensing decisions.

#### A. Ground truth model for target detection uncertainty

Let  $\xi_i \in \{0, 1\}$  denote the output of an object detector with perfect detection ability which labels an object  $i$  at a distance  $l_i$  away from the camera accurately with either a ‘0’ (not OOI) or a ‘1’ (OOI). Typically, the confidence score of the object detector gradually declines as a function of the OOI’s distance from the camera. Prior work [16] has characterized this behavior using a depth aware detection model where the output  $y_i$  of an imperfect object detector is the true object label modified by an additive one-sided Gaussian noise:

$$y_i = \xi_i + n_i^d, \text{ with } n_i^d \sim \mathcal{N}^+(0, \sigma_i^2(l_i)). \quad (1)$$

Therefore,  $y_i \in [0, 1]$  follows a one-sided normal distribution centered at  $\xi_i=0$  for a true negative and  $\xi_i=1$  for a true positive label. The variance  $\sigma_i^2(\cdot)$  is an increasing function of  $l_i$ . So the probability of a false negative or a false positive OOI detection increases at distances farther away from the camera. This is illustrated in Fig. 1c where the horizontal axis indicates the detector’s  $y_i$  and the curves indicate the varying probability densities at different object distances  $l_i$ .

#### B. Ground truth model for target location uncertainty

Let  $\zeta_i \in \mathbb{R}$  denote the measurement from an accurate depth sensor of its true distance to an object  $i$ . Real depth

measurements have error along the agent’s line of sight. Prior work [17] has characterized the error as a Gaussian distribution. In our setup, we model the target location uncertainty as additive Gaussian noise so that

$$y_i = \zeta_i + n_i^\ell, \text{ with } n_i^\ell \sim \mathcal{N}(0, r_u^2) \quad (2)$$

where  $y_i \in \mathbb{R}$  is the measurement from an imperfect depth sensor and  $r_u$  parameterizes the uncertainty of such measurements along the agent’s line of sight. As depicted in Fig. 1d, the measured depth follows a normal probability distribution centered on the true target location.

#### C. Problem Formulation

We consider a gridded search environment described by a sparse matrix which we want to recover through multi-agent active search.  $M$  is the total number of grid cells.  $\beta \in \{0, 1\}^{M \times 1}$  denotes the flattened vector representation of the environment having  $k$  non-zero entries at the true locations of the  $k$  OOIs.  $\mathbf{X}_t \in \{0, 1\}^{Q_t \times M}$  is the sensing action at time  $t$ .  $Q_t$  is the number of grid cells covered by the robot’s FOV under  $\mathbf{X}_t$ . Each row of  $\mathbf{X}_t$  is a one hot vector  $\{0, 1\}^{1 \times M}$  indicating the position of one of the sensed grid cells in the robot’s FOV.  $\mathbf{X}_t \beta \in \{0, 1\}^{Q_t \times 1}$  is the ground truth observation due to sensing action  $\mathbf{X}_t$ .  $\mathbf{y}_t \in \mathbb{R}^{Q_t \times 1}$  is the agent’s observation vector indicating the noisy sensor measurement from executing  $\mathbf{X}_t$ . The sensing model is

$$\underbrace{\mathbf{y}_t}_{\text{noisy measurement}} = \underbrace{\mathbf{X}_t \beta}_{\text{ground truth observation}} + \underbrace{\mathbf{n}_t}_{\text{measurement noise}} \quad (3)$$

where  $\mathbf{n}_t \in \mathbb{R}^{Q_t \times 1}$  is composed of the noise from (1), (2).

**Remark 1.** Note that the model described above is what our simulator considers to be ground truth. Our algorithm and its agents are neither aware of the number of targets nor their true locations, and only receive the measurement  $(\mathbf{X}_t, \mathbf{y}_t)$ .

**Communication:** We assume that communication, although unreliable, will be available sometimes and the agents should communicate when possible. The agents share their measurements asynchronously, and do not wait on inter-agent communications. Further, the set of available past measurements need not remain consistent across agents due to communication unreliability.

**Remark 2.** Since our goal is uncertainty-aware active sensing and not planning a continuous path, we only require a coarse discretization of our environment. For example, in Section V-C, we cover a 250m×250m region with square grid cells of size 15m. We also assume that individual OOIs occupy an entire grid cell and the location uncertainty from the depth sensor only affects which cell was determined to have the OOI, not the OOI’s placement within the cell.

To recover the search vector  $\beta$  by actively identifying all the OOIs, at each time  $t$ , an agent  $j$  chooses the best sensing action  $\mathbf{X}_t^j$  based on its belief about the OOIs given the measurements available thus far in its measurement set  $\mathbf{D}_t^j$ . Assuming that all the agents collectively obtain  $T$  measurements, our objective is to correctly estimate the sparse vector  $\beta$  with as few measurements  $T$  as possible. For a single agent, our problem reduces to sequential decision making with

the measurement set  $\mathbf{D}_t^1 = \{(\mathbf{X}_1, \mathbf{y}_1), \dots, (\mathbf{X}_{t-1}, \mathbf{y}_{t-1})\}$  available to the agent at time  $t$ . In the multi-agent setting, following our communication constraints, we use a decentralized and asynchronous parallel approach with agents independently deciding individual sensing actions [18], [16], [19].

### III. RELATED WORK

The issue of location uncertainty has been studied by the robotics community, particularly in tasks such as path planning, localization and tracking [20], [21]. Such problems are typically approached using filtering algorithms that can recursively estimate the robot or target state and the associated location uncertainty, while abstracting away the detection uncertainty by thresholding the detection probability to zero or one. Sensor measurements from RGB cameras and depth sensors used in typical robotic tasks like SLAM and navigation often introduce data association error due to pose mismatch between the corresponding sensors [22]. Further, depth images suffer from position dependent geometric distortions and distance dependent measurement bias [17]. This also introduces location uncertainty in the sensor measurements. As a result, sensor calibration algorithms have been proposed which parameterize the error in the depth measurements and then learn those parameters from carefully collected training data [23], [24], [25], [26], [27]. But they typically engineer away the need to account for detection uncertainty by using some predetermined visual patterns in the collected dataset. Unlike our problem setup, they do not focus on learning how to autonomously collect appropriate data to reduce location uncertainty in the sensor measurements.

The domain of sparse signal recovery, on the other hand, is primarily concerned with the problem of choosing sensing actions in the face of detection uncertainty, assuming that once the signal is detected then its location is accurate. Prior work in this area has proposed algorithms that use principles from compressed sensing together with constrained optimization to estimate the signal [28], [29]. [30] formulated this as an active search problem with realistic region sensing actions and [31] proposed a reinforcement learning approach incorporating the detection uncertainty in the observations into a POMDP framework. [19], [16] build on the former setup and go on to model detection uncertainty as a function of the target's distance from the sensing agent.

The discrete spatial search problem is also studied in search theory by considering detection uncertainty through false positive detections in sensing individual cells [32], [33], [34] but they do not relate the uncertainty with sensor capabilities and do not generalize to realistic region sensing actions.

In terms of the agent's decision making, active search has close similarities with the task of information gathering for planning and navigation in robotics. The majority of such algorithms use entropy and information gain as an optimization objective for deciding which action to execute [10], [30]. Unfortunately this leads to deterministic action selection for an information greedy agent. As a result, much of the work in this domain has generally focused on single agent settings [35], [36], [37], and if multi-agent, they typically

rely on the presence of a centralized controller to coordinate the agents so that they can avoid all choosing the same action at any decision making step [38], [39]. Recently, motivated by the real world constraints of centralized multi-agent systems, some algorithms have been proposed with a centralized planning followed by decentralized execution approach [40], [41], [42]. Further, some others propose to achieve decentralization by having the agents repeatedly communicate their future plans (e.g. beliefs or intended sensing actions) with their peers [43], [44]. This can become cumbersome especially if the belief is high dimensional and there is unreliable inter-agent communication. In contrast, [19], [16] consider a realistic setup and introduce posterior sampling based active search in a decentralized multi-agent system with unreliable at-will communication of sensing measurements among peers.

The task of characterizing the uncertainty of detection and location of objects in an image has also been extensively studied in computer vision [45], [46], [47], [48], [49]. In contrast, our agents adaptively *choose* which images to capture (i.e. decide where to sense and in which direction) by considering the associated uncertainties to recover all OOIs in a search space.

### IV. OUR PROPOSED ALGORITHMS: UNIK AND TS-UNIK

We now describe our approach to the multi-agent active search problem described in Section II-C in two stages. First, we outline our inference procedure that agents use to identify OOIs and estimate their locations given the set of available measurements thus far. Next, we describe our decentralized and asynchronous multi-agent decision making algorithm that utilizes the estimates from the proposed inference method.

#### A. Proposed inference algorithm: Unik

Following Section II-C, we want to recover  $\beta$  by identifying all the OOIs. But both the number and position of targets are unknown to the agents. So we initialize each agent with a Gaussian prior over  $\beta$ , denoted by  $p_0(\beta) = \mathcal{N}(\hat{\beta}_0, \mathbf{P}_0)$ . Given the measurement set  $\mathbf{D}_t^j$  available to the agent  $j$  at time  $t$ , it updates its posterior belief  $p(\beta|\mathbf{D}_t^j) = \mathcal{N}(\hat{\beta}_t^j, \mathbf{P}_t^j)$  using a Kalman filter (KF) [50] with process model  $\beta_t = \beta_{t-1}$  and measurement model  $\mathbf{z}_t = \mathbf{X}_t\beta_t + \nu_t$ , where  $\nu_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{z}_t})$ . The prediction and update steps of the KF are as follows.

$$\text{Prediction: } \hat{\beta}_t^- = \hat{\beta}_{t-1}, \mathbf{P}_t^- = \mathbf{P}_{t-1} \quad (4)$$

$$\text{Kalman gain: } \mathbf{K}_t = \mathbf{P}_t^- \mathbf{X}_t^T (\mathbf{X}_t \mathbf{P}_t^- \mathbf{X}_t^T + \Sigma_{\mathbf{z}_t} + \lambda_t \mathbf{I})^{-1} \quad (5)$$

$$\text{Update: } \hat{\beta}_t = \hat{\beta}_t^- + \mathbf{K}_t (\mathbf{y}_t - \mathbf{X}_t \hat{\beta}_t^-) \quad (6)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{X}_t) \mathbf{P}_t^- (\mathbf{I} - \mathbf{K}_t \mathbf{X}_t)^T + \mathbf{K}_t \Sigma_{\mathbf{z}_t} \mathbf{K}_t^T \quad (7)$$

$\lambda_t$  is a regularization constant. Given sensing action  $\mathbf{X}_t$  and observation  $\mathbf{y}_t$  (3), the agent constructs measurement noise covariance matrix  $\Sigma_{\mathbf{z}_t}$  to incorporate both target detection and location uncertainty in the KF measurement model. First,  $\Sigma_{\mathbf{z}_t} = \text{diag}([\sigma_q^2(l_q)]_{q=1}^{Q_t})$  is initialized to account for the OOI detection uncertainty in the robot's FOV using the distance dependent variance (Section II-A). Next, the agent estimates

the possible OOI locations in its FOV by thresholding  $\mathbf{y}_t$ :  $\mathbf{y}_{thr,t} = \mathbf{1}(\mathbf{y}_t \geq c_{thr})$ . For each possible OOI in  $\mathbf{y}_{thr,t}$ , the agent then determines its location uncertainty field i.e. the grid cells in its FOV where this OOI might be truly located.

Following Fig. 2a, suppose  $\mathbf{y}_{thr,t}$  indicates an OOI in the position marked “X” in red, whereas the OOI is actually located in the grid cell marked “X” in black. Along its line of sight to the observed OOI location (“X” in red), the agent computes the location uncertainty field using the detector’s radial uncertainty parameter  $r_u$  (same as (2)) and an angular uncertainty parameter  $\theta_u$  shown in Fig. 2a. It will include all the grid cells in the region bounded radially by  $[-r_u, r_u]$  around the observed OOI location within an angular spread of  $[-\theta_u, \theta_u]$  relative to the line of sight to the agent. Next, the location uncertainty field for each possible OOI index in  $\mathbf{y}_{thr,t}$  is used to compute the variance and covariance due to location uncertainty in  $\Sigma_{\mathbf{z}_t}$ .

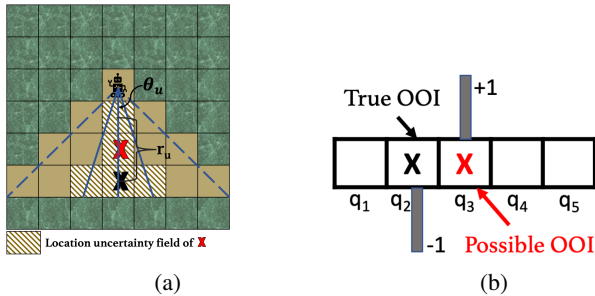


Fig. 2: (a) Showing the location uncertainty field for an OOI detected in  $\mathbf{y}_{thr,t}$  at the position marked “X” in red. (b) Showing part of the additive noise vector due to target location uncertainty in an agent’s observation (3) when an OOI is actually present at  $q_2$  but observed at  $q_3$ .

TABLE I: Modeling the probability distribution for target location uncertainty in the measurement model using the observed target location at  $q_3$  and its uncertainty field  $\{q_1, q_2, q_3, q_4, q_5\}$ .

$\tilde{q}_{OOI}$	$q_{OOI}$	$\mathbf{n}_{t,q_{OOI}}^l$
$q_3$	$q_1$	[ -1 0 1 0 0 ... 0] <sup>T</sup>
	$q_2$	[ 0 -1 1 0 0 ... 0] <sup>T</sup>
	$q_3$	[ 0 0 0 0 0 ... 0] <sup>T</sup>
	$q_4$	[ 0 0 0 1 -1 0 ... 0] <sup>T</sup>
	$q_5$	[ 0 0 0 1 0 -1 ... 0] <sup>T</sup>

Using the computed location uncertainty field for an OOI at  $\tilde{q}_{OOI}$  in its FOV, the agent computes a probability function over the grid cells where the OOI could truly be located. Referring to the observation noise due to location uncertainty  $\mathbf{n}_t^l$  in Section II-B, we note that in a discretized grid environment, it essentially results in swapping the OOI’s observed location within a region around the true OOI location. This needs to be captured in the measurement model for the agent’s KF. For example, following Fig. 2b, if  $\mathbf{y}_{thr,t}$  indicates that an OOI is located at  $\tilde{q}_{OOI} = q_3$  and its location uncertainty field is  $\{q_1, q_2, q_3, q_4, q_5\}$ , Table I shows

the possible location swaps the agent reasons about including the case where the OOI’s true location is also  $q_3$ . In Table I,  $\tilde{q}_{OOI}$  is the observed location,  $q_{OOI}$  is the possible true location and  $\mathbf{n}_{t,q_{OOI}}^l$  is the noise component due to location uncertainty that executes the position swap between the true and observed OOI locations. The index with  $-1$  indicates  $q_{OOI}$  and that with  $1$  indicates  $\tilde{q}_{OOI}$ . After enumerating all feasible OOI position swaps in the location uncertainty field, the agent updates the measurement noise covariance as  $\Sigma_{\mathbf{z}_t}^{q_3, q_3} += 0^2 \times \frac{1}{5} + (1)^2 \times (1 - \frac{1}{5})$  and  $\Sigma_{\mathbf{z}_t}^{q_1/2/4/5, q_3} = \Sigma_{\mathbf{z}_t}^{q_3, q_1/2/4/5} += 1 \times (-1) \times \frac{1}{5} + 1 \times 0 \times \frac{3}{5} + 0 \times 0 \times \frac{1}{5}$ . This is repeated for all possible OOIs in the agent’s FOV. The constructed measurement noise covariance  $\Sigma_{\mathbf{z}_t}$  is then used to update the agent’s KF with  $(\mathbf{X}_t, \mathbf{y}_t)$ .

#### Algorithm 1 UnIK (inference with random action selection)

- 1: **Assume:** Sensing model (3), sparse true state  $\beta$
- 2:  $\mathbf{D}_0^j = \phi$ ,  $p_0^j(\beta) = \mathcal{N}(\hat{\beta}_0^j, \mathbf{P}_0^j)$ ,  $\hat{\beta}_0^j = \frac{1_{M \times 1}}{M}$ ,  $\mathbf{P}_0^j = \sigma_0^2 \mathbf{I}$
- 3: **for**  $t$  in  $\{1, \dots, T\}$  **do**
- 4:   Select  $\mathbf{X}_t^j$  uniform randomly; observe  $\mathbf{y}_t^j$ .
- 5:   Update  $\mathbf{D}_t^j = \mathbf{D}_{t-1}^j \cup \{(\mathbf{X}_t^j, \mathbf{y}_t^j)\}$
- 6:   Obtain  $\mathbf{y}_{thr,t}^j = \mathbf{1}(\mathbf{y}_t^j \geq c_{thr})$
- 7:   Initialize  $\Sigma_{\mathbf{z}_t}$  to account for detection uncertainty
- 8:   **for** each possible OOI in  $\mathbf{y}_{thr,t}^j$  **do** (Section IV-A)
- 9:     Obtain its location uncertainty field; update  $\Sigma_{\mathbf{z}_t}$
- 10:   Estimate  $\hat{\beta}_t^j, \mathbf{P}_t^j$  using KF

#### B. Proposed action selection algorithm: TS-UnIK

In order to develop an action selection strategy for our agents, we observe that our active search formulation falls within the problem setup for the Myopic Posterior Sampling (MPS) framework developed in [51]. Like MPS, our objective is to actively learn  $\beta$  with as few measurements as possible. MPS chooses the action that maximizes the one-step lookahead (myopic) reward assuming that a sample drawn from the posterior is the true state of the world. This idea is a generalization of Thompson sampling (TS) [52] which is commonly used in Bayesian optimization. Furthermore, parallelized asynchronous TS is an excellent candidate for a decentralized multi-agent algorithm [18], [16], [19]. By using a posterior sample in the reward function, TS (as well as MPS) enables multiple agents to independently choose distinct sensing actions that maximize their respective rewards while incurring little regret compared to a centralized planner. Therefore we adopt a TS approach in our proposed algorithm.

Next we describe our TS based reward formulation. At time  $t$ , an agent  $j$  having available measurements  $\mathbf{D}_{t-1}^j$  estimates its posterior belief  $p(\beta | \mathbf{D}_t^j) = \mathcal{N}(\hat{\beta}_{t-1}^j, \mathbf{P}_{t-1}^j)$  using UnIK. It then draws a sample  $\tilde{\beta}_t^j \sim p(\beta | \mathbf{D}_t^j)$ . Assuming  $\tilde{\beta}_t^j$  to be the true search vector, the agent will prefer choosing an action  $\mathbf{X}_t^j$  that will allow its updated estimate  $\hat{\beta}_t^j$  at time  $t$  to be as close as possible to  $\tilde{\beta}_t^j$ . In particular, we want to maximize

$$\mathcal{R}(\tilde{\beta}_t^j, \mathbf{D}_{t-1}^j, \mathbf{X}_t^j) = \frac{\mathbb{E}_{\mathbf{z}_t^j | \tilde{\beta}_t^j, \mathbf{X}_t^j}[-\|\tilde{\beta}_t^j - \hat{\beta}_t^j\|^2]}{\mathbb{E}_{\mathbf{z}_t^j | \tilde{\beta}_t^j, \mathbf{X}_t^j}[\|\hat{\beta}_t^j\|^2]}$$

where  $\mathbf{z}_t^j$  is the KF measurement variable. Note that  $\hat{\beta}_t^j$  is a one-step lookahead estimate assuming that  $\mathbf{X}_t^j$  would result in an observation  $\mathbf{z}_t^j$  following the KF measurement model if  $\beta_t^j = \hat{\beta}_t^j$ . The numerator favours actions  $\mathbf{X}_t^j$  such that in expectation, the estimate  $\hat{\beta}_t^j$  is close to the assumed true state  $\tilde{\beta}_t^j$ . The denominator  $\mathbb{E}_{\mathbf{z}_t^j|\hat{\beta}_t^j, \mathbf{X}_t^j} [\|\hat{\beta}_t^j\|_2^2]$  is analogous to the estimated strength of the signal, and since the numerator is non-positive, a larger denominator will lead to an overall higher reward. Therefore, our formulated reward prefers sensing actions that would maximally reduce the uncertainty in the agent's current posterior distribution.

Using  $\hat{\beta}_t^j = \hat{\beta}_{t-1}^j + \mathbf{K}_t^j(\mathbf{z}_t^j - \mathbf{X}_t^j\hat{\beta}_{t-1}^j)$  and  $\mathbf{z}_t^j \sim \mathcal{N}(\mathbf{X}_t^j\hat{\beta}_t^j, \Sigma_{\mathbf{z}_t^j})$ , we can compute  $\mathcal{R}(\hat{\beta}_t^j, \mathbf{D}_{t-1}^j, \mathbf{X}_t^j)$  as:

$$\begin{aligned} & -\|\hat{\beta}_t^j - \hat{\beta}_{t-1}^j + \mathbf{K}_t^j\mathbf{X}_t^j\hat{\beta}_{t-1}^j\|_2^2 + 2(\tilde{\beta}_t^j - \hat{\beta}_{t-1}^j \\ & \quad + \mathbf{K}_t^j\mathbf{X}_t^j\hat{\beta}_{t-1}^j)^T\mathbf{K}_t^j\mathbf{X}_t^j\tilde{\beta}_t^j - \|\mathbf{K}_t^j\|_2^2(\text{tr}(\Sigma_{\mathbf{z}_t^j}) + \|\mathbf{X}_t^j\hat{\beta}_t^j\|_2^2) \\ & \frac{\|\hat{\beta}_{t-1}^j - \mathbf{K}_t^j\mathbf{X}_t^j\hat{\beta}_{t-1}^j\|_2^2 + \|\mathbf{K}_t^j\|_2^2(\text{tr}(\Sigma_{\mathbf{z}_t^j}) + \|\mathbf{X}_t^j\tilde{\beta}_t^j\|_2^2)}{\|\hat{\beta}_{t-1}^j - \mathbf{K}_t^j\mathbf{X}_t^j\hat{\beta}_{t-1}^j\|_2^2 + \|\mathbf{K}_t^j\|_2^2(\text{tr}(\Sigma_{\mathbf{z}_t^j}) + \|\mathbf{X}_t^j\tilde{\beta}_t^j\|_2^2)} \\ & \quad + 2(\hat{\beta}_{t-1}^j - \mathbf{K}_t^j\mathbf{X}_t^j\tilde{\beta}_t^j)^T\mathbf{K}_t^j\mathbf{X}_t^j\tilde{\beta}_t^j \end{aligned} \quad (8)$$

Finally, among all actions  $\{\mathbf{X}_t^j\}$  at time  $t$ , agent  $j$  chooses  $\mathbf{X}_t^{j*}|\hat{\beta}_t^j = \arg \max_{\mathbf{X}_t^j} \mathcal{R}(\hat{\beta}_t^j, \mathbf{D}_{t-1}^j, \mathbf{X}_t^j)$ .

#### Algorithm 2 TS-UnIK

- 1: **Assume:**Sensing model (3), sparse true state  $\beta$ ,  $J$  agents
- 2:  $\mathbf{D}_0^j = \phi$ ,  $p_0^j(\beta) = \mathcal{N}(\hat{\beta}_0^j, \mathbf{P}_0^j)$ ,  $\hat{\beta}_0^j = \frac{1}{M} \mathbf{1}_{M \times 1}$ ,  $\mathbf{P}_0^j = \sigma_0^2 \mathbf{I}$
- 3: **for**  $t$  in  $\{1, \dots, T\}$  **do** ▷ For any available agent  $j$
- 4:   Sample  $\hat{\beta}_t^j \sim p(\beta|\mathbf{D}_{t-1}^j) = \mathcal{N}(\hat{\beta}_{t-1}^j, \mathbf{P}_{t-1}^j)$
- 5:    $\mathbf{X}_t^{j*} = \arg \max_{\mathbf{X}_t^j} \mathcal{R}(\hat{\beta}_t^j, \mathbf{D}_{t-1}^j, \mathbf{X}_t^j)$ . Observe  $\mathbf{y}_t^j$ .
- 6:   Update  $\mathbf{D}_t^j = \mathbf{D}_{t-1}^j \cup \{(\mathbf{X}_t^{j*}, \mathbf{y}_t^j)\}$ . Share  $(\mathbf{X}_t^{j*}, \mathbf{y}_t^j)$  asynchronously with teammates.
- 7:   Estimate  $\hat{\beta}_t^j, \mathbf{P}_t^j$  using UnIK ( Line 6-Line 10)

**Remark 3.** TS based decision making enables each agent to independently choose its next sensing action based on the uncertainty in its current posterior over the search space, and subsequently update its individual posterior estimates using its own measurements and those from other agents. We do not require perfect data association between measurements from different agents on the same OOI, nor a central controller for synchronization of observations across agents.

**Computational complexity :** Due to the Kalman filter based recursive nature of UnIK, for any agent, the time complexity of each inference step is the same and bounded by  $O(M^{2.376})$ . Additionally, TS-UnIK requires an agent to select the best (maximum reward) among all feasible actions at each time step. Therefore Line 5 in Algorithm 2 results in  $O(|\mathcal{A}|M^2)$  complexity where  $|\mathcal{A}|$  is the size of the agent's action space.

## V. EXPERIMENTAL RESULTS

We now describe the simulation setup for our experimental results. Consider a 2-dimensional (2D) discretized search space having  $J$  agents tasked with recovering the unknown search vector  $\beta$  which is randomly generated using a uniform sparse prior with  $k$  non-zero entries each with value 1. Note

that the agents are not aware of the number of targets (OOIs)  $k$ . We assume that agents are positioned at the centre of the grid cells they occupy and are free to move in any direction in the search space. When positioned in a cell, an agent can look in one of 4 possible directions: north, south, east or west to a maximum distance of 5 grid cells ahead with a 90° FOV.

In the following experiments, the agents' performance is measured using the full recovery rate which is defined as the rate at which the agents correctly recover the *entire* vector  $\beta$ . The plots show mean values with shaded regions indicating standard error over multiple trials, each trial differing only in the instantiation of the true position of the  $k$  OOIs in  $\beta$ .

### A. Comparing inference

We compare UnIK against two other inference methods: (1) NATS [16], which only accounts for detection uncertainty and (2) LU, which is our designed location uncertainty only baseline. Fig. 3 illustrates the performance of UnIK compared to NATS and LU in a single agent setting ( $J = 1$ ) in a  $16 \times 16$  search space, over 50 random trials. At each step, an action is chosen uniformly at random. The agent's actions and corresponding observations are the same across the three inference algorithms.

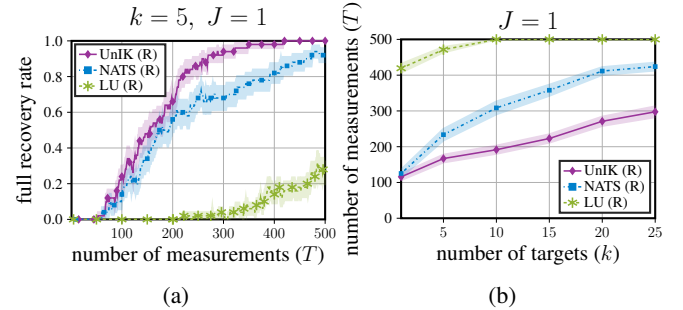


Fig. 3:  $J = 1$  agent in a  $16 \times 16$  grid,  $k$  targets. Mean and standard error over 50 trials. (R) indicates uniformly random action selection. UnIK outperforms baseline inference algorithms that account for only one of detection and location uncertainty, for different number of targets.

Fig. 3a shows the full recovery rate of an agent when there are  $k=5$  targets in a  $16 \times 16$  search space. Unlike NATS or LU, UnIK leverages the combined target detection and location uncertainty to detect all the OOIs with fewer measurements. Fig. 3b further compares the efficiency of the algorithms in terms of number of measurements needed to achieve full recovery by varying the number of targets  $k \in \{1, 5, 10, 15, 20, 25\}$ . With increasing  $k$ , NATS lacking the target location uncertainty modeling requires an increasingly larger number of measurements compared to UnIK. On the other hand, in the absence of target detection uncertainty modeling, LU's performance is sensitive to the detection threshold and declines faster, failing in all 50 trials to recover all OOIs within  $T = 500$  measurements for  $k > 5$ .

## B. Comparing action selection

We now demonstrate the efficiency of TS based action selection in combination with our proposed inference method for decentralized and asynchronous multi-agent active search. **UnIK (R) vs. TS-UnIK** : Following our observations in Section V-A where UnIK outperformed other inference algorithms in fully recovering all OOIs, we will now use UnIK with uniformly random action selection (i.e. UnIK (R)) as the baseline against which we compare the performance of TS-UnIK. Fig. 4a plots the number of measurements needed

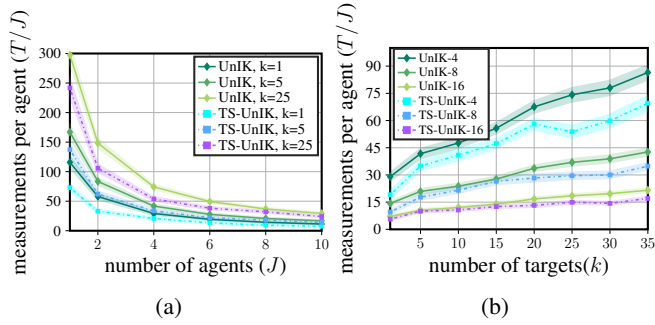


Fig. 4: TS-UnIK scales better with number of agents  $J$  and targets  $k$  compared to UnIK (R) in a  $16 \times 16$  grid, 50 trials.

per agent ( $T/J$ ) to fully recover  $k$  OOIs as the number of agents  $J$  increases. In a perfect sensing setup, we would expect a single agent algorithm using the TS based action selection strategy to require  $J$  times as many measurements as that required per agent in a team with  $J$  agents [19]. We observe this to hold for TS-UnIK, resulting in a  $J$  times improvement in performance when the number of agents multiplies  $J$  times. As the team size becomes larger, we observe that the full recovery performance plateaus in the absence of inter-agent coordination or centralized control.

Fig. 4b shows the robustness of multi-agent active search with TS-UnIK in terms of the number of measurements per agent required by a team to fully recover all OOIs as the number of OOIs  $k$  increases. For different team sizes ( $J$ ) (indicated as UnIK/TS-UnIK- $J$ ), for both UnIK (R) and TS-UnIK to fully recover all OOIs, the average number of measurements needed per agent in the team increases with more number of OOIs in the search space. However in all the settings, TS-UnIK outperforms UnIK (R) and the performance gap widens further with more targets  $k$  for different  $J$ .

**TS-UnIK vs. NATS** : We now compare the performance of TS-UnIK against NATS with TS based action selection [16]. Fig. 5a shows the full recovery rate within  $T = 150$  measurements in a single agent setting ( $J = 1$ ) when there is  $k = 1$  target in a  $16 \times 16$  search space. In this setting, TS-UnIK outperforms NATS and enables the single agent to achieve the desired recovery rate more efficiently with significantly fewer measurements. Fig. 5b further validates the superiority of TS-UnIK over NATS in a multi-agent setting with multiple OOIs. For different team sizes  $J \in \{4, 8, 16\}$  with the same total number of measurements  $T = 500$ , we compare the full recovery rate achieved by the team in terms of the number

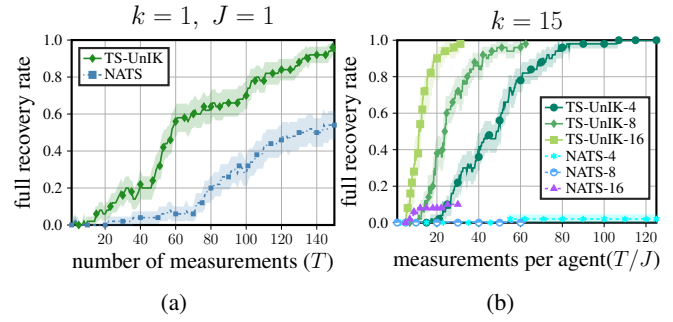


Fig. 5:  $J$  agents in a  $16 \times 16$  grid with  $k$  targets, 50 trials. TS-based action selection benefits more from the joint uncertainty modeling in UnIK, than with NATS [16]. The performance improvement is particularly significant for multi-agent teams recovering multiple targets in the search space.

of measurements required per agent. As before, we observe that TS-UnIK is able to efficiently recover all the OOIs and demonstrates a  $J$  times improvement in performance as the team size grows by a factor of  $J$ . On the other hand, with an identical team size and the same number of measurements per agent, NATS fails to achieve full recovery in the majority of trials. The deterioration in performance due to the absence of target location uncertainty modeling in NATS's inference method as observed in Section V-A, is further exacerbated in the decentralized multi-agent setting with asynchronous communication. In particular, when different agents perform overlapping sensing actions, they may observe the same OOI at different locations due to observation noise arising from target location uncertainty. As a result, the combined detection and location uncertainty modeling in UnIK not only helps better estimate the possible OOI positions but this improved posterior belief also helps TS choose effective sensing actions, leading to faster recovery of all OOIs with TS-UnIK.

## C. TS-UnIK in Unreal Engine 4 (UE4) with AirSim plugin

We test TS-UnIK in a pseudo-realistic environment created in UE4 with an AirSim plugin. It is a  $250m \times 250m$  field with trees and animals, discretized into  $16 \times 16$  grid cells. There are  $k = 5$  humans randomly positioned within the field, who are the OOIs for our ground robot. We use YOLOv3 [15] (off-the-shelf) as the agent's object detector, which provides a label and confidence score for OOIs in the agent's FOV (Fig. 1b). Using the depth maps provided by AirSim, we implement our own depth sensor model that provides noisy location measurements to the agent following Eq. (2). Using measurements from these sensors, the agent decides its next sensing action following Eq. (8), with an additional travel distance penalization term. We have included a video demonstrating the performance of TS-UnIK in this environment.<sup>1</sup> Supporting the results in Section V-B, the agent successfully recovers all OOIs with sensing actions that decide where and how the surroundings are observed to eliminate false or missed detections over time.

<sup>1</sup>Also at <https://sites.google.com/view/unik-icra23>.

## REFERENCES

- [1] R. Garnett, Y. Krishnamurthy, D. Wang, J. Schneider, and R. Mann, "Bayesian optimal active search on graphs," in *Ninth Workshop on Mining and Learning with Graphs*. Citeseer, 2011.
- [2] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. Mann, "Bayesian optimal active search and surveying," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 843–850.
- [3] R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.
- [4] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, 1997, pp. 193–200.
- [5] G. Flaspohler, V. Preston, A. P. Michel, Y. Girdhar, and N. Roy, "Information-Guided Robotic Maximum Seek-and-Sample in Partially Observable Continuous Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3782–3789, 2019.
- [6] A. Alsayed, A. Yunusa-Kaltungo, M. K. Quinn, F. Arvin, and M. R. Nabawy, "Drone-Assisted Confined Space Inspection and Stockpile Volume Estimation," *Remote Sensing*, vol. 13, no. 17, p. 3356, 2021.
- [7] B. Huang, J. Zhao, and J. Liu, "A survey of simultaneous localization and mapping," *arXiv preprint arXiv:1909.05214*, 2019.
- [8] J. Haupt, R. Nowak, and R. Castro, "Adaptive Sensing for Sparse Signal Recovery," in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, 2009, pp. 702–707.
- [9] M. L. Malloy and R. D. Nowak, "Near-Optimal Adaptive Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4001–4012, 2014.
- [10] P. Rajan, W. Han, R. Sznitman, P. Frazier, and B. Jedynak, "Bayesian multiple target localization," in *International conference on machine learning*. PMLR, 2015, pp. 1945–1953.
- [11] R. Marchant and F. Ramos, "Bayesian optimisation for Intelligent Environmental Monitoring," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2242–2249.
- [12] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari, "Online-to-confidence-set conversions and application to sparse stochastic bandits," in *Artificial Intelligence and Statistics*. PMLR, 2012, pp. 1–9.
- [13] A. Carpentier and R. Munos, "Bandit theory meets compressed sensing for high dimensional stochastic linear bandit," in *Artificial Intelligence and Statistics*. PMLR, 2012, pp. 190–198.
- [14] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.
- [15] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
- [16] R. Ghods, W. J. Durkin, and J. Schneider, "Multi-agent active search using realistic depth-aware noise model," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9101–9108.
- [17] A. Belhedi, A. Bartoli, S. Bourgeois, K. Hamrouni, P. Sayd, and V. Gay-Bellile, "Noise modelling and uncertainty propagation for tof sensors," in *European Conference on Computer Vision*. Springer, 2012, pp. 476–485.
- [18] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos, "Parallelised bayesian optimisation via thompson sampling," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 133–142.
- [19] R. Ghods, A. Banerjee, and J. Schneider, "Decentralized multi-agent active search for sparse signals," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 696–706.
- [20] V. Caglioti, A. Citterio, and A. Fossati, "Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach," in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, 2006, pp. 25–30.
- [21] A. Stroupe, M. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, 2001, pp. 1092–1098 vol.2.
- [22] F. Basso, E. Menegatti, and A. Pretto, "Robust intrinsic and extrinsic calibration of RGB-D cameras," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1315–1332, 2018.
- [23] Q.-Y. Zhou and V. Koltun, "Simultaneous localization and calibration: Self-calibration of consumer depth cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 454–460.
- [24] D. Zuñiga-Noël, J.-R. Ruiz-Sarmiento, and J. Gonzalez-Jimenez, "Intrinsic calibration of depth cameras for mobile robots using a radial laser scanner," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2019, pp. 659–671.
- [25] G. Chen, G. Cui, Z. Jin, F. Wu, and X. Chen, "Accurate Intrinsic and Extrinsic Calibration of RGB-D Cameras With GP-Based Depth Correction," *IEEE Sensors Journal*, vol. 19, no. 7, pp. 2685–2694, 2019.
- [26] S. Miller, A. Teichman, and S. Thrun, "Unsupervised extrinsic calibration of depth sensors in dynamic scenes," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2695–2702.
- [27] F. Basso, A. Pretto, and E. Menegatti, "Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6244–6249.
- [28] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for Sparse Signal Recovery Using Kalman Filtering With Embedded Pseudo-Measurement Norms and Quasi-Norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, 2010.
- [29] D. Needell and R. Vershynin, "Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, 2010.
- [30] Y. Ma, R. Garnett, and J. Schneider, "Active search for sparse signals with region sensing," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [31] C. Igoe, R. Ghods, and J. Schneider, "Multi-Agent Active Search: A Reinforcement Learning Approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 754–761, 2022.
- [32] M. Kress, K. Y. Lin, and R. Szechtman, "Optimal discrete search with imperfect specificity," *Mathematical methods of operations research*, vol. 68, no. 3, pp. 539–549, 2008.
- [33] T. H. Chung and J. W. Burdick, "Analysis of Search Decision Making Using Probabilistic Search Strategies," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 132–144, 2012.
- [34] T. Cheng, B. Krieheli, E. Levner, and C. Ng, "Scheduling an autonomous robot searching for hidden targets," *Annals of Operations Research*, pp. 1–15, 2019.
- [35] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Robotics: Science and Systems*, 2015.
- [36] Z. W. Lim, D. Hsu, and W. S. Lee, "Adaptive informative path planning in metric spaces," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 585–598, 2016.
- [37] T. Patten, W. Martens, and R. Fitch, "Monte Carlo planning for active object classification," *Autonomous Robots*, vol. 42, no. 2, pp. 391–421, 2018.
- [38] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong, "Multimodal active perception for information gathering in science missions," *Autonomous Robots*, vol. 43, no. 7, pp. 1827–1853, 2019.
- [39] H. Surmann, R. Worst, T. Buschmann, A. Leinweber, A. Schmitz, G. Senkowski, and N. Goddemeier, "Integration of uavs in urban search and rescue missions," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2019, pp. 203–209.
- [40] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 66–83.
- [41] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.
- [42] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2961–2970.
- [43] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "DecMCTS: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

- [44] G. Best and G. A. Hollinger, "Decentralised self-organising maps for multi-robot information gathering," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4790–4797.
- [45] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Suenderhauf, "Probabilistic Object Detection: Definition and Evaluation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.
- [46] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 680–688.
- [47] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3022–3031.
- [48] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2488–2496.
- [49] Z. Wang, D. Feng, Y. Zhou, L. Rosenbaum, F. Timm, K. Dietmayer, M. Tomizuka, and W. Zhan, "Inferring Spatial Uncertainty in Object Detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5792–5799.
- [50] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [51] K. Kandasamy, W. Neiswanger, R. Zhang, A. Krishnamurthy, J. Schneider, and B. Poczos, "Myopic posterior sampling for adaptive goal oriented design of experiments," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3222–3232.
- [52] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.