Active localization of multiple targets from noisy relative measurements

Selim Engin and Volkan Isler

Department of Computer Science and Engineering
University of Minnesota, Minneapolis, MN 55455
{engin003, isler}@umn.edu

Abstract. Consider a mobile robot tasked with localizing targets at unknown locations by obtaining relative measurements. The observations can be bearing or range measurements. How should the robot move so as to localize the targets and minimize the uncertainty in their locations as quickly as possible? This is a difficult optimization problem for which existing approaches are either greedy in nature or they rely on accurate initial estimates.

We formulate this path planning problem as an unsupervised learning problem where the measurements are aggregated using a Bayesian histogram filter. The robot learns to minimize the total uncertainty of each target in the shortest amount of time using the current measurement and an aggregate representation of the current belief state. We analyze our method in a series of experiments where we show that our method outperforms a standard greedy approach. In addition, its performance is comparable to that of an offline algorithm which has access to the true location of the targets. [1]

1 Introduction

Environmental monitoring is an application area where robotics can have a major impact. Robots can be used for gathering data, collecting samples and performing surveillance across large environments over long periods of time. A practical problem of interest is the localization of animals in the wild who have been previously radio-tagged by wildlife biologists. The animals can be localized by mobile robots which can obtain bearing measurements using a directional antenna. Figure 1 shows two examples of such platforms built by our group.

In this paper, we study the problem of localizing multiple targets given noisy measurements of their locations relative to a robot. In particular, we study the bearing-only and range-only measurement models which are commonly encountered in practice. For both of these models, the mathematical form of the localization uncertainty given the true target location and robot measurement locations is well established (Section 3.1). However, computing the measurement locations is hard when the true location of the target is unknown or if there are multiple targets.

---

[1] More information available at https://ksengin.github.io/active-target-localization/

Fig. 1: **The target localization problem:** given noisy relative measurements, the goal is to localize the targets as quickly as possible using a robot, possibly with complex dynamics such as a quadrotor (left) or surface vehicle [1] (right).

Existing methods described in Section 2 mostly focused on the case where there is only a single target. Moreover, most of them are suboptimal greedy strategies, optimizing the mutual information or entropy at each step locally. This approach is however susceptible to getting stuck in local minima when the best next steps are in conflicting directions. Other approaches rely on accurate initial estimates, which may not be available in general settings. We show that instead of using a local greedy approach, long-term planning strategies which maximize the expected rewards can help the localization system. We assume all the targets to be initially observable, therefore we do not address the search problem. However, a noisy initial observation of the target can result in arbitrarily bad initial estimates. Our method is able to accurately localize targets using a bearing-only sensor with noise level up to $30°$ angles, and range measurements with up to 30 meters of error in a $200 \times 200 \ m^2$ area.

Our work lies in the class of model-free Reinforcement Learning (RL), where the robot learns to accomplish a given task in a fixed time-horizon without using prior information about the environment or robot dynamics. In contrast to model-based methods, this approach does not use an engineered or a learned dynamics model. Instead, it directly optimizes its policy through a sequence of trial-and-error process [2].

While the policy learning is performed in a model-free fashion, our method uses a representation of the environment that encapsulates the sensor modality and the uncertainty of the estimations. Specifically, we use a top-down image of the environment where the intensity of each pixel indicates the likelihood of that point being the position of the target. As the sensor noise increases there are more pixels indicating a high likelihood of containing the target.

2   Related Work and Contributions

The problem of localizing targets has received significant attention and been studied in several different settings. In this section we go over three different scenarios where each setup has a different set of assumptions about the target and sensor mobility: a) sensor placement, the problem of deploying a group of static sensors to localize static targets; b) active target localization, the problem

of localizing static targets with mobile sensors; and c) active target tracking, the problem of tracking mobile targets with mobile sensors.

2.1 Sensor Placement

In the sensor placement problem we have a group of static targets whose positions are unknown residing in a given candidate area and an uncertainty model. The goal is usually to minimize the number of sensors deployed in the area given a desired uncertainty level [3–5], or to optimize the sensor-target geometries after fixing the number of sensors [6]. Both the targets and sensors are assumed to be static in this setting, and a greedy approach is shown to have a near-optimal performance [3]. The results in this setting, however, are not directly applicable to the case when the targets or sensors are moving.

2.2 Active Target Localization

In the problem of active target localization, one or multiple mobile robots need to localize the targets as quickly as possible. In [7] a method is proposed to perform target localization with bearing measurements using the Fisher information to optimize the trajectory of a single aerial vehicle. A cooperative algorithm is presented in [1] to localize a static target using two robots with communication constraints. The robots in this work need to periodically meet to update their estimate of the target position. [8] studied the case where an unknown number of targets are localized with a team of robots. It also addresses the problem of data association, that is the measurements do not have unique identifiers.

2.3 Active Target Tracking

In the active target tracking problem, both the sensors and targets are able to move. The target motion can be modeled in various ways and this model plays an important role in the performance of the tracking system. One approach is to treat the target motion adversarial, which means that the target actively evades the sensor [9,10]. Another approach can model the motion of the target to be a random walk, especially when there is no prior information about the mobility of the target [11,12]. The mobility is modeled by a 2D Brownian motion in [13] where the target can move within a bounded region.

For this problem setting, most of the existing approaches use a position-based measurement model [14–17]. This enables the robot to estimate the target positions with a few observations, unlike using bearing or range measurements where the geometry of the sensor-target configuration significantly impacts the accuracy of the estimator.

2.4 Statement of contributions

Our contributions can be summarized as follows.

– We propose a novel RL formulation of the multi-target localization problem.
– We present a representation to model the localization estimates for each target encapsulating the uncertainty in the predictions.
– We compare our RL algorithm against a greedy approach that uses local observations to plan the sensor motion, and an offline algorithm which has access to the true positions of the targets.

3   Preliminaries

The setup in our problem consists of a tracker robot with an omnidirectional bearing or range sensor and a set of m targets deployed in an environment $\mathcal{V}$. The mobility of the robot is modeled to be holonomic, that is it can do instantaneous turns without any kinematic constraints. We denote the position of the robot at time t by $p_t \in \mathbb{R}^2$. Similarly, the locations of the targets are $\mathbf{q} = \{q_1, \ldots, q_m\}$, where $q_i \in \mathbb{R}^2$, and the predicted positions of the targets are $\hat{\mathbf{q}}$. The trajectory of the robot until time step t is denoted by $P_{1:t} = p_1, \ldots, p_t$. Throughout the paper we use a global frame, such as the initial coordinate frame of the robot to denote the positions of the robot and targets.

The true bearing direction from a sensor location $p = (p^x, p^y)$ to a target location $q_i = (q_i^x, q_i^y)$ is denoted by $\phi_i = \phi(p, q_i)$ and expressed as:

$$\phi_i = \arctan\left((q_i^y - p^y)/(q_i^x - p^x)\right). \tag{1}$$

We assume that the measurement model has a Gaussian distribution noise. Then, a bearing measurement from a sensor can be written as $\hat{\phi}_i = \phi_i + e$, where $e \sim \mathcal{N}(0, \sigma_s^2)$ and σ_s^2 is the variance of the sensor measurement error. We denote the set of bearing measurements acquired from a robot position p_t at time step t by $\hat{\Phi}_t = \{\hat{\phi}_1, \ldots \hat{\phi}_m\}$.

The distance between a sensor p and a target q_i is denoted by $d_i = d(p, q_i)$. Similar to the bearing measurement model, we assume a normally distributed zero-mean noise for the range measurements. A range measurement at time t is expressed as $\hat{d}_i = d_i + e$, where $e \sim \mathcal{N}(0, \sigma_s^2)$. Throughout the paper we denote the relative noisy measurements by $\hat{z}_i$, and use $\hat{\phi}_i$ or $\hat{d}_i$ to indicate a specific sensor model. The set of relative measurements at time t is denoted by $\hat{\mathbf{z}}_t$.

3.1   Uncertainty Measures

In this section we describe the uncertainty measures used in the paper.

Fisher Information Predicting a target location using relative measurements is typically achieved by triangulation from multiple positions. Quantifying the accuracy of the triangulation relies on the geometry of the sensor-target configuration and the noise in the sensor measurement.

For a given sensor-target geometry and a noise level in the measurements, we can use the Fisher Information Matrix (FIM) to compute the amount of

information given by the measurements. The FIM characterizes the amount of information an observable parameter carries about an unobservable variable. In our case, the observable parameter is the relative measurement and the unobservable variable of interest is the position of the target.

Suppose $\mathcal{F}_q$ is the FIM for a given sensor-target configuration, where q^* is the true location of the target. The (i, j)-th entry of $\mathcal{F}_q$ for a single target can be expressed as:

$$(\mathcal{F}_q)_{i,j} = \mathbb{E}\Big[\frac{\partial}{\partial q_i} \ln(\mathbb{P}(\hat{z}|q)) \ \frac{\partial}{\partial q_j} \ln(\mathbb{P}(\hat{z}|q))\Big], \tag{2}$$

where $\mathbb{P}(\cdot)$ denotes the probability density function. The determinant of the FIM is inversely proportional to the uncertainty area of the estimation [18, 19]. The determinant of $\mathcal{F}_q$ for n bearing measurements is given by [20]:

$$det(\mathcal{F}_q) = \frac{1}{\sigma_s^4} \sum_{\{p_i, p_j\} \in \mathcal{P}} \frac{\sin^2(\phi(p_i, q^*) - \phi(p_j, q^*))}{d(p_i, q^*)^2 \cdot d(p_j, q^*)^2}, \ \ i < j \tag{3}$$

where σ_s^2 denotes the variance of the measurement error, and $\mathcal{P}$ defines the set of all pairwise combinations as $\{\{p_i, p_j\}\}$ with $i, j \in \{1, \ldots, n\}$ and $i < j$. The inverse FIM $\mathcal{F}_q^{-1}$ (also known as the Cramer-Rao inequality lower bound) quantifies an uncertainty ellipsoid for unbiased estimators. The square root of the eigenvalues of $\mathcal{F}_q^{-1}$, denoted by $\sqrt{\lambda_i}$ defines the axis length of the ellipsoid along the i-th dimension.

The determinant of $\mathcal{F}_q^{-1}$ provides a scalar for measuring the uncertainty. For a given sensor-target geometry and a measurement model, we measure the total uncertainty by:

$$\mathcal{U}(P, \mathbf{q}) = \sum_{i=1}^{m} det(\mathcal{F}_{q_i}^{-1}). \tag{4}$$

Since the true position of the target is required to compute the Fisher information, it is challenging to obtain an accurate uncertainty estimate based on the Fisher information in practical scenarios. The next uncertainty measure we present is better suited for practical applications where we do not have access to the true positions. We use the Fisher information as an uncertainty estimate in the offline case where the true position of the targets are available to the algorithm.

Bayesian Histograms In Bayesian histograms we discretize the environment $\mathcal{V}$ into a grid, where each cell of the grid indicates the likelihood of being the target position given the measurements. Bayesian histograms are particularly useful to represent the likelihood posteriors for models with nonlinear dynamics. Filtering techniques relying on linearization, in general are sensitive to disturbances when using a nonlinear sensor model like bearing measurements. Without

accurate initialization, the measurement errors can easily lead to a poor localization performance. Several existing methods [21, 22] have also used Bayesian histograms for the problem of target localization.

Since the measurement model assumes a zero-mean normal distribution for the sensor noise, we can compute the probability of a point v being the true target location q^* given a relative measurement $\hat{z}$ as:

$$\mathbb{P}(v = q^*|p, \hat{z}) = f(z_v|\hat{z}, \sigma_s^2) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\{-\frac{1}{2\sigma_s^2}(z_v - \hat{z})^2\}, \tag{5}$$

where z_v is the true angle $\phi(p, v)$ or distance $d(p, v)$ between the point v and a robot position p. Over the course of the robot's trajectory, the likelihood of a point is updated as measurements are obtained:

$$\mathbb{P}(v = q^*|P_{1:T}, \hat{z}_{1:T}) = \prod_{k=1}^{T} f(z(p_k, v)|\hat{z}_k, \sigma_s^2). \tag{6}$$

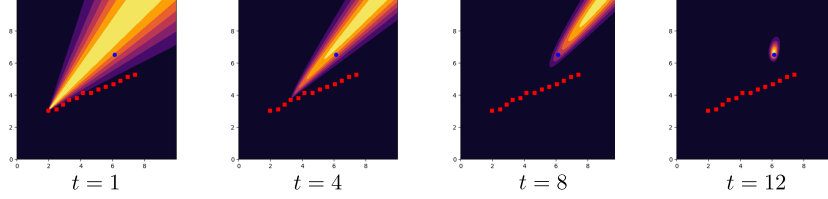$$t = 1 \qquad\qquad t = 4 \qquad\qquad t = 8 \qquad\qquad t = 12$$

Fig. 2: The Bayesian histograms computed by aggregating the bearing measurements taken from the sensor trajectory shown in red. The blue dot corresponds to the true target location, and the darker to brighter colors in the heatmap indicate an increase in the likelihood.

At each measurement update, the probabilities are normalized such that $\max_{v \in \mathcal{V}} \mathbb{P}(v = q^*|P_{1:T}, \hat{z}_{1:T}) = 1$, where $\mathcal{V}$ is a given domain for the target positions. We use the max operation to represent relative probabilities per each bin in the histogram and for numerical stability. When initializing the heatmap, we use a uniform distribution of probability 1 indicating that each point in the domain is equally likely to be the true position of the target. We denote the histogram at time step t for a target q_i as $\mathcal{H}_t(P_{1:t}, \hat{\mathbf{z}}_{1:t}, i)$. The histograms are updated separately and stacked as a tensor $\mathcal{H}_t(P_{1:t}, \hat{\mathbf{z}}_{1:t}) \in \mathbb{R}^{m \times W \times H}$. The histogram dimensions $W \times H$ determine the granularity of the grid discretization of the environment and result in a trade-off between accuracy and efficiency. Finer resolution histograms have smaller discretization errors at the cost of higher computational resources. An instance of the Bayesian histogram representation is shown in Figure 2. At each time step the bearing measurement is incorporated to update the likelihood of the points in $\mathcal{V}$. The sensor trajectory and target position are overlaid in the heatmap figures to give context.

3.2   Reinforcement Learning

In Reinforcement Learning (RL), we have an agent sensing and acting within an environment over a course of discrete time steps. At each time step, the agent observes the current state of the environment and performs an action. Depending on the current state and action pair, the state changes and the agent receives a reward according to an environment model [2].

These type of problems are usually formulated as Markov Decision Processes (MDP) described by states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition dynamics $\mathcal{T}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$, a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and a discount factor $\gamma \in [0,1]$. For the case when the agent cannot fully observe the states, this description is redefined as a Partially Observable MDP (POMDP), where there is also an associated observation function.

The agent selects actions using a policy $\pi(a|s)$ so as to maximize the expected γ-discounted future rewards given by $G_t = \mathbb{E}\big[\sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i, s_{i+1})\big]$, where T is the planning time-horizon. Suppose that $Q^\pi(s, a)$ is the expected future returns by taking the action a in state s using the policy π, which is expressed as $Q^\pi(s, a) = \mathbb{E}\big[G_t | s_t = s, a_t = a\big]$. The goal of RL algorithms is to learn an optimal policy π^* that maximizes the expected future returns for all state and action pairs. A popular method for this problem is Q-learning [23], where the Bellman equation,

$$Q(s_t, a_t) = \mathbb{E}_{\mathcal{T}(s_{t+1}|s_t, a_t)}\big[r(s_t, a_t, s_{t+1}) + \gamma \max_{a'} Q(s_{t+1}, a')\big], \qquad (7)$$

is recursively used to adjust Q^π to approach the optimal Q-values for π^*. When the state and action spaces are large, this value function can be approximated as a neural network $Q_\theta(s, a)$, with parameters θ.

4   Problem Statement

We are now ready to formally state the problem studied in this paper.

Problem 1 *(Target Localization)*. We are given a mobile robot equipped with a sensor obtaining relative measurements according to a known measurement model, a starting position p_1, a fixed planning horizon T and a set of targets whose positions are unknown. We assume that each target transmits a unique signal, thus there is no correspondence problem for the measurements. In addition, we are given a function $\mathcal{U}(P, q)$ measuring the uncertainty of the target estimates for a given sensor-target geometry and a measurement model. The goal is to find a path $P_{1:T} = p_1, \ldots, p_T$ for the robot such that the total uncertainty of the target positions $\sum_{i=1}^{m} \mathcal{U}(P_{1:T}, q_i)$ is minimized.

5   Method

Our method, what we call Active Target Localizer (ATL), uses two different models to represent the state. In this section we go over these representations and present our unsupervised learning method for target localization.
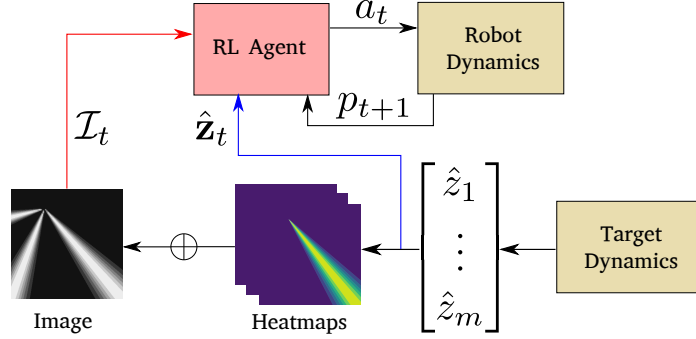
Fig. 3: **Method overview:** A single iteration of the robot-environment inter-action. The blue arrow indicates the relative measurements vector used in the multi-modal representation, while the red arrow marks the image used as the observation in the image representation.

An overview of our method is shown in Figure 3. At every iteration, the robot acquires noisy measurements from the target locations according to the measurement model. The measurements are either taken as raw observations, or they are aggregated into a single image representation (shown in blue and red arrows, respectively). Depending on the observation, the agent selects an action which drives the robot to a new position. The measurements are accumulated in the form of heatmaps at each step, which are then used to predict the position of the target. This process repeats until the end of the planning time-horizon.

5.1   Representing the observations

We start with describing the representations we use for the target observations.

Multi-modal representation The first representation our method uses is a multi-modal Gaussian distribution for characterizing the localization belief. At each step, the robot receives a distinct bearing or range measurement from all the targets (i.e. perfect assignment). The sensor noise σ_s is assumed to be known, therefore we are able to represent the likelihood distribution over the domain $\mathcal{V}$. We construct the likelihood of each point $v \in \mathcal{V}$ as the normalized probability $f(z_v|\hat{z}, \sigma_s^2)$, where $\hat{z}$ is the observation for a single target.

In a standard Markov process, at any time step the next state depends on the current state and the action. For localizing the targets, the robot needs to maintain an estimate of the target position using observations from the past locations. Hence, we append the predictions for the target positions to the state vector so as to select the next actions.

The augmented state of the system is defined as $s_t := (p_t, \hat{\mathbf{z}}_t, \hat{\mathbf{q}}_t) \in \mathbb{R}^{2+m+2m}$, concatenating the position vector of the robot p_t, observations from each target $\hat{\mathbf{z}}_t$, and a flattened vector of the predicted positions of the targets $\hat{\mathbf{q}}_t$. In the

multi-modal representation, we use the noisy bearing $\hat{\phi}_t$ or range measurements $\hat{d}_t$ as the observation vector $\hat{\mathbf{z}}_t$.

Image representation We also use images to represent the belief for the predicted position of the target. The image $\mathcal{I}_t \in \mathbb{R}^{W \times H}$ at time step t is computed by the sum of histograms maintained for each target estimate, expressed as:

$$\mathcal{I}_t = \sum_{i=1}^{m} \mathcal{H}_t(P_{1:t}, \hat{\mathbf{z}}_{1:t}, i). \tag{8}$$

The image is then normalized to have values in the range $[0, 1]$, and resized into a desired resolution with bilinear interpolation. We use a convolutional neural network with the ResNet-34 architecture [24] followed by three fully connected layers to encode $\mathcal{I}_t$ into a latent vector $l_t \in \mathbb{R}^c$. For all our experiments, we set the dimension of the latent code to be $c = 128$. This encoding vector implicitly describes the uncertainty level and the position estimates for all the targets in a latent space.

The state vector we use in the image representation is expressed as $s_t := (p_t, l_t) \in \mathbb{R}^{2+c}$, a concatenation of the position vector p_t and latent code l_t. The RL algorithm we present next inputs the state vector corresponding to the desired observation representation.

5.2   Learning target localization

We build our method on the Twin Delayed Deep Deterministic (TD3) algorithm, a deep reinforcement learning algorithm introduced in [25] that uses policy gradients to directly optimize the policy network. Similar to the Deep Deterministic Policy Gradient algorithm [26], TD3 uses an actor-critic architecture that jointly learns an actor π_φ (the policy) and a critic function Q_θ (the value function) modeled by neural networks with parameters φ and θ, respectively.

A replay buffer is maintained to store the previous interactions of the agent, as in the Deep Q-Network algorithm [27]. The replay buffer stores samples in the form of $(s_i, a_i, s_{i+1}, r_i, y_i)$ tuples, where each sample is associated with a state transition from s_i to s_{i+1} by executing the action a_i. With this transition, the agent receives a reward r_i, and the termination of the episode is indicated by a label $y_i \in \{0, 1\}$. To prevent the over-estimation bias of the value function, TD3 trains two (twin) critic networks that are updated at different (delayed) frequencies and uses the minimum of the two as the value estimate. In our notation, $\min_{j=1,2} Q_{\theta_j}(s, a)$ is replaced by $Q_\theta(s, a)$ for brevity.

For both the actor and critic, periodically updated target networks $\pi_{\varphi'}$ and $Q_{\theta'}$ are also used. Constraining the change in the target networks to be slow helps the stabilization when learning the actor and critic functions.

The actor and critic networks are trained with the Adam optimizer [28] using the loss functions L_{actor} and L_{critic} over mini-batches of size B. The loss

functions are given by:

$$L_{actor} = \frac{1}{B} \sum_i^B -Q_\theta(s_i, \pi_\varphi(s_i)) \qquad (9)$$

$$L_{critic} = \frac{1}{B} \sum_i^B \left[Q_\theta(s_i, a_i) - \left(r_i + \gamma(1 - y_i) \cdot Q_{\theta'}(s_{i+1}, \pi_{\varphi'}(s_{i+1})) \right) \right]^2. \qquad (10)$$

Intuitively, the actor network π_φ is trained to maximize the Q-values generated from the action $\pi_\varphi(s_i)$ given a state s_i, by minimizing $-Q_\theta(s_i, \pi_\varphi(s_i))$. On the other hand, the critic network is trained to minimize the Bellman error computed by a variation of the Bellman equation given in Equation (7). During the training process, the slowly changing target networks are assumed to characterize the optimal policy and the Q-function. The policy π_φ inputs the state vector s_t according to the observation representation and outputs an action a_t. This action is a heading angle in the continuous space represented in the inertial coordinate frame, taking values bounded as $a_t \in [0, 2\pi)$.

Reward functions For learning the policy we use two different reward functions, one for each observation representation. In the multi-modal representation, the reward is the negative mean squared error between the true and predicted target positions, expressed as $-1/m \cdot \sum_i^m ||\mathbf{q}_i - \hat{\mathbf{q}}_i||^2$.

For the image representation, on the other hand, we simply use the mean of the image intensities as the loss term. The reward function we use is given by $-\frac{1}{WH} \cdot \sum_{u,v} \mathcal{I}_t(u, v)$, where the pixel intensities of the image are all non-negative. Maximizing this reward function minimizes the likelihood of parts of the environment $\mathcal{V}$ not containing the targets, which in turn amounts to minimizing the total entropy of the Bayesian histograms. The reward function does not use the true position of the targets, thus it is unsupervised.

6   Analysis

In this section we evaluate our method ATL with a set of experiments. The experiments are designed around questions concerned about the performance of our method compared to several baseline approaches.

6.1   Offline Trajectory Optimization

We first investigate whether we can generate optimal trajectories for a robot to minimize the uncertainty, given the true position of the targets.

This problem has been addressed in two recent papers for bearing [29] and range-only [30] measurement models, where they study the optimal one-step action for the sensor so as to maximize the information about the target estimate. The cost function they provide for one-step maximization has the form:

$J_\phi = \sin(\phi(p_{t-1}, q) - \phi(p_t, q))^2 / d(p_t, q)^2$, which is used to select the best action to move the sensor from position p_{t-1} to p_t when using bearing measurements. This expression closely resembles the geometric dilution of precision (GDOP) [31], commonly used in navigation systems. The uncertainty estimate using the GDOP function for a given sensor-target geometry is given by:

$$U_{GDOP} = \frac{d(p_i, q) \cdot d(p_j, q)}{|\sin(\phi(p_i, q) - \phi(p_j, q))|}. \tag{11}$$

The offline algorithm we compare against is based on the minimization of the uncertainty measure $\mathcal{U}(P, \mathbf{q})$. This measure is equal to a constant times the GDOP function summed over all pairs of the measurement locations P. We call this offline algorithm OFFLINE-FISHER and present it in Algorithm 1.

Algorithm 1 OFFLINE-FISHER

Input: $\mathcal{U}$: uncertainty function , $\mathbf{q}$: target positions, T: time horizon, δ_p: step size
Output: $P_{1:T}$: a trajectory for minimizing the uncertainty of the target estimates
1: $p_1 \leftarrow$ Starting position of the robot
2: **for** $t \leftarrow 2$ to T **do**
3:     $a_t^* \leftarrow \arg\min_{a_t \in \mathcal{A}} \sum_{i=1}^m \mathcal{U}(P_{1:t-1} \cup \{p_t\}, q_i)$
4:     $p_t^* \leftarrow \delta_p \cdot \big[\cos(a_t^*); \sin(a_t^*)\big]; P_{1:t} \leftarrow P_{1:t-1} \cup \{p_t^*\}$
5: **return** $P_{1:T}$

6.2 Localizing Static Targets

In practical settings, the true location of the targets are not available to us, therefore the algorithm needs to operate under uncertainty. The second method we compare against is a local greedy approach which chooses an action to minimize the conditional entropy given the past measurements of the targets, at each step. This approach is commonly used in practice and it resembles e.g. [32].

In this section we present experimental evaluations comparing our method against OFFLINE-FISHER and the local greedy algorithms. Specifically, we investigate the contribution of having expected future rewards in the cost function, in contrast to optimizing only the next step.

The environment used in our experiments is a 20×20 square area. The robot step size is set to be 0.5 units, and the planning time-horizon is 50 steps. Variance of the sensor noise for bearing measurements is $\sigma_s^2 = 0.2^2$ which corresponds to about 11°, and for range measurements it is 1. Each unit in the histogram has a resolution of 10, thus the dimensions of $\mathcal{H}_t$ are 200×200.

Localization of two targets with bearing and range-only measurements using the image representation is shown in Figure 4. The figure shows the measurements acquired from the targets by the robot, and its trajectory to minimize the uncertainty of the predictions by using the corresponding images. We see that the learned policy is able to generate smooth trajectories that can accurately localize multiple targets.

(a) Target localization with bearing measurements

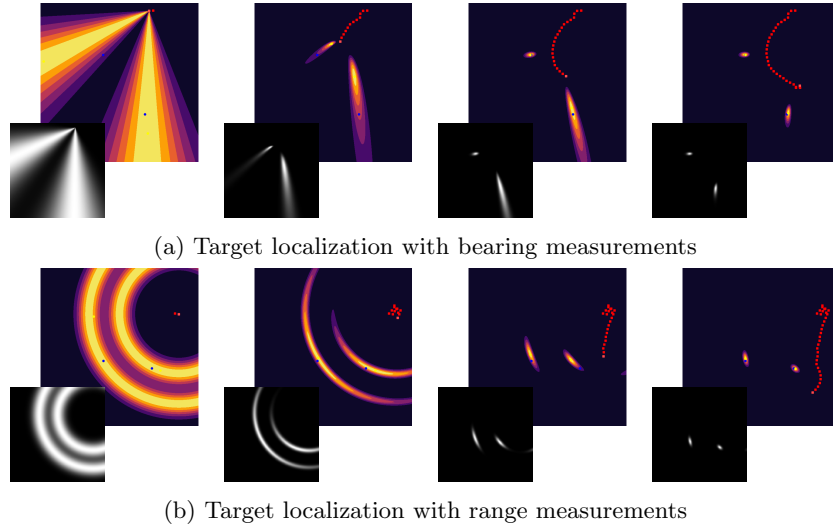

(b) Target localization with range measurements

Fig. 4: The trajectory of the robot acquiring bearing (a) and range (b) measurements, using the image representation. Both rows show the robot trajectory (red), true (blue) and predicted (yellow) target position, also the Bayesian histogram as the colored heatmap. The insets contain the images taken as input to the network at corresponding instants. Best viewed in color.

We provide quantitative results for the localizing the targets using bearing and range-only measurements in Figure 5. The plots show the mean localization error using four methods, averaged over 100 episodes. We compute the localization error by finding the point with the highest likelihood from the uncertainty histograms, and use it to calculate the mean distance error between the predicted ($\hat{\mathbf{q}}$) and true ($\mathbf{q}$) target positions. We observe that both representations we use outperform the greedy strategy over the course of an episode, and have a performance very close to that of the offline algorithm.

The mean localization error at the end of the episodes for each method is presented in Table 1. Unsurprisingly, the best trajectories minimizing the localization uncertainty are generated by the offline algorithm, in all cases. We see that our method using either of the observation representation yields very close results to the offline algorithm. The performance of the two representations are similar using bearing measurements, however the multi-modal representation results in better localization when using range measurements. One possible cause for this is the spatial characteristics of the image in the case of range measurements. In bearing measurements the centroid of the high-likelihood pixels gives a reasonable prediction about the target position, which is not always the case for range measurements. This phenomenon could also be the reason why OFFLINE-FISHER outperforms our method ATL by a larger margin when using range measurements. However, in all experiments we see that our method outperforms the greedy strategy.

(a) $m = 4$, bearing (ϕ) (b) $m = 8$, bearing (ϕ)

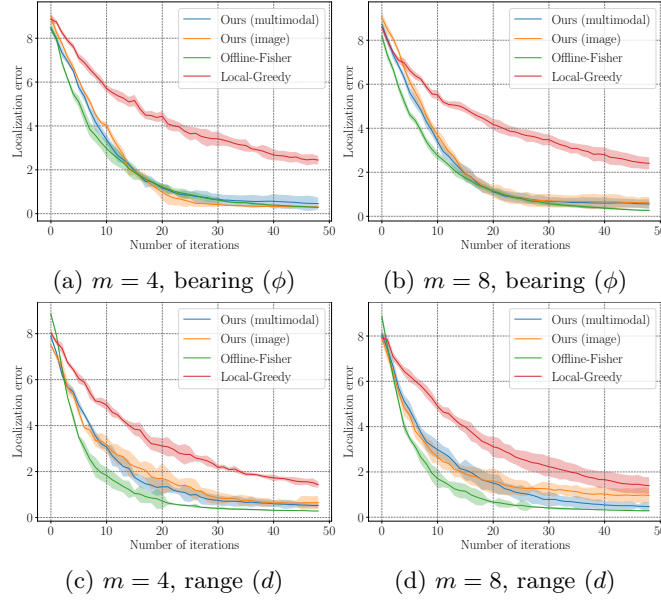(c) $m = 4$, range (d) (d) $m = 8$, range (d)

Fig. 5: Mean localization errors computed over 100 trials using bearing and range measurements from $m = 4, 8$ targets. Shaded area represents 1σ.

	Method	$m = 2$	$m = 4$	$m = 8$
ϕ	Offline-Fisher	*0.228 / 0.012*	*0.285 / 0.006*	*0.268 / 0.301*
	Greedy-Local	2.599 / 0.404	2.444 / 0.190	2.406 / 0.284
	ATL (Multi-modal)	0.481 / 0.092	0.457 / 0.272	**0.557 / 0.223**
	ATL (Image)	**0.479 / 0.255**	**0.305 / 0.041**	0.618 / 0.256
d	Offline-Fisher	*0.276 / 0.001*	*0.282 / 0.007*	*0.284 / 0.016*
	Greedy-Local	1.357 / 0.264	1.430 / 0.137	1.388 / 0.381
	ATL (Multi-modal)	**0.461 / 0.070**	**0.524 / 0.104**	**0.468 / 0.186**
	ATL (Image)	0.990 / 0.257	0.646 / 0.299	0.954 / 0.336

Table 1: Mean and standard deviation of localization errors using bearing (ϕ) and range (d) measurements from static targets in a 20×20 area

6.3 Localizing Dynamic Targets

We next analyze the localization performance when the targets are moving locally in a contained area. The target movement is modeled by a 2D Brownian motion with a covariance of $0.1I_2$, similar to the setting in [13]. Our current formulation of the uncertainty histogram does not allow tracking targets performing large displacements. However, when the target motion is relatively small our method gives reasonable performances compared to the baseline algorithms.

The results of this experiment are reported in Table 2. We see that the error dynamics follow a similar trend to the case of stationary targets. Moreover, for

all the methods the localization performance degrades as the targets start to move.

	Method	$m = 2$	$m = 4$	$m = 8$
	Offline-Fisher	*0.696 / 0.042*	*0.707 / 0.023*	*0.706 / 0.039*
ϕ	Greedy-Local	3.177 / 0.396	2.851 / 0.305	2.810 / 0.257
	ATL (Multi-modal)	0.981 / 0.661	0.992 / 0.728	**0.899 / 0.444**
	ATL (Image)	**0.901 / 0.215**	**0.699 / 0.140**	1.031 / 0.368
	Offline-Fisher	*0.591 / 0.028*	*0.597 / 0.013*	*0.611 / 0.033*
d	Greedy-Local	1.985 / 0.106	2.003 / 0.149	2.159 / 0.338
	ATL (Multi-modal)	**0.947 / 0.034**	**0.874 / 0.134**	**0.845 / 0.180**
	ATL (Image)	1.312 / 0.328	0.978 / 0.279	1.390 / 0.195

Table 2: Localization errors using relative measurements from moving targets

6.4   Generalization Performance

The experiments reported so far evaluated the networks with the same number of targets they have seen during training. In this section we investigate the generalization performance of our method when there are more number of targets than what the network was trained on.

Since the image representation is agnostic to the number of targets to be localized, testing our networks on generalized number of targets is seamless. In this set of experiments, we take our policy network trained with 2 targets and evaluate it on $m = 6$ and 12 static targets. The results presented in Table 3 suggest that our method generalizes to localizing unseen number of targets, owing to the image representation it uses for the observations.

Method	$m = 6$	$m = 12$
Offline-Fisher	*0.315 / 0.067*	*0.286 / 0.038*
Greedy-Local	2.316 / 0.146	1.921 / 0.299
ATL (Image)	**0.729 / 0.202**	**0.728 / 0.267**

Table 3: Generalization performance: Evaluation on $m = 6, 12$ targets using a policy network trained to localize 2 targets with bearing measurements

7   Conclusion

In this paper we presented a method for localizing multiple targets using bearing or range measurements. Leveraging recent advances in reinforcement learning and convolutional networks, our method is able to generate trajectories for a robot to localize targets when obtaining noisy relative observations. We show that our method outperforms a standard greedy approach and performs similarly compared to an offline algorithm which has access to the true position of the targets, without knowing the true target locations.

Future work includes tracking multiple targets that have adversarial movements. Another interesting research direction is to study the target localization problem in higher dimensions and cluttered environments such as animal tracking with an aerial vehicle in a forest.

Acknowledgement

References

1. Joshua Vander Hook, Pratap Tokekar, and Volkan Isler. Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints. *IEEE Transactions on Robotics*, 31(4):864–876, 2015.
2. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.
3. Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 265–272. ACM, 2005.
4. Onur Tekdas and Volkan Isler. Sensor placement for triangulation-based localization. *Transactions on Automation Science and Engineering*, 7(3):681–685, 2010.
5. Krishnendu Chakrabarty, S Sitharama Iyengar, Hairong Qi, and Eungchun Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers*, 51(12):1448–1453, 2002.
6. Shiyu Zhao, Ben M Chen, and Tong H Lee. Optimal placement of bearing-only sensors for target localization. In *2012 American Control Conference (ACC)*, pages 5108–5113. IEEE, 2012.
7. Sameera Ponda, Richard Kolacinski, and Emilio Frazzoli. Trajectory optimization for target localization using small unmanned aerial vehicles. In *AIAA Guidance, Navigation, and Control Conference*, page 6015, 2009.
8. Philip Dames and Vijay Kumar. Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. *IEEE Transactions on Automation Science and Engineering*, 12(3):850–864, 2015.
9. Simone Battistini and Tal Shima. Differential games missile guidance with bearings-only measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):2906–2915, 2014.
10. Tal Shima and Josef Shinar. Time-varying linear pursuit-evasion game models with bounded controls. *J. of Guidance, Control, and Dynamics*, 25(3):425–432, 2002.
11. Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Int. Conf. on Robotics and Automation*, pages 1665–1670, 2001.
12. Ting Yu and Ying Wu. Decentralized multiple target tracking using netted collaborative autonomous trackers. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 939–946. IEEE, 2005.
13. Haruki Nishimura and Mac Schwager. Sacbp: Belief space planning for continuous-time dynamical systems via stochastic sequential action control. In *The 13th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

14. Heejin Jeong, Brent Schlotfeldt, Hamed Hassani, Manfred Morari, Daniel D Lee, and George J Pappas. Learning q-network for active information acquisition. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
15. Ruijie He, Abraham Bachrach, and Nicholas Roy. Efficient planning under uncertainty for a target-tracking micro-aerial vehicle. In *International Conference on Robotics and Automation*, pages 1–8. IEEE, 2010.
16. Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13):1660–1677, 2015.
17. Alyssa Pierson and Daniela Rus. Distributed target tracking in cluttered environments with guaranteed collision avoidance. In *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 83–89. IEEE, 2017.
18. Harry L Van Trees. *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*. John Wiley & Sons, 2004.
19. Adrian N Bishop, Baris Fidan, Brian DO Anderson, Kutluyil Dogancay, and Pubudu N Pathirana. Optimality analysis of sensor-target geometries in passive localization: Part 1-bearing-only localization. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 7–12. IEEE, 2007.
20. Adrian N Bishop, Barış Fidan, Brian DO Anderson, Kutluyıl Doğançay, and Pubudu N Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.
21. Folker Hoffmann, Hans Schily, Alexander Charlish, Matthew Ritchie, and Hugh Griffiths. A rollout based path planner for emitter localization. In *International Conference on Information Fusion (FUSION)*, pages 1–8, 2019.
22. Oliver M Cliff, Robert Fitch, Salah Sukkarieh, Debra L Saunders, and Robert Heinsohn. Online localization of radio-tagged wildlife with an autonomous aerial robot system. In *Robotics: Science and Systems*, 2015.
23. Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
24. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
25. Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.
26. Timothy P Lillicrap, Jonathan J Hunt, Pritzel, et al. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
27. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
28. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
29. Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Trajectory optimization for target localization with bearing-only measurement. *Transactions on Robotics*, 35(3):653–668, 2019.
30. Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Optimal active target localisation strategy with range-only measurements. In *Int. Conf. on Informatics in Control, Automation and Robotics*, 2019.
31. Alonzo Kelly. Precision dilution in triangulation based mobile robot position estimation. In *Intelligent Autonomous Systems*, volume 8, pages 1046–1053, 2003.
32. Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Robotics research*, pages 455–472. Springer, 2017.