Mutual Information Maps for Single and Multi-Target Ergodic Search

Thesis by

Howard Coffin

In Partial Fulfillment of the Requirements for the
Degree of
Masters of Science in Robotics

CARNEGIE MELLON UNIVERSITY
Pittsburgh, Pennsylvania

2021
Defended 6 August 2021

© 2021

Howard Coffin
ORCID: 0000-0001-5256-8456

All rights reserved

ACKNOWLEDGEMENTS

I would like to thank my mentors Mike Greene and Chris Fairley, without whom I would never have studied robotics.

ABSTRACT

Though robots have become more prevalent in search and rescue operations, they usually require human operators to direct their search. Automating the search process can allow larger teams of robots to be used in a wider variety of situations, such as when communication is limited. Furthermore, automated robots have the potential to outperform teleoperated ones, especially in cases where it is difficult for a human to interpret incoming sensor data in real time.

Recent works have used ergodic search methods to automatically generate search trajectories. Ergodic search scales to a larger number of agents compared to typical information-based algorithms while still allowing prior knowledge to be incorporated into the search procedure. The prior knowledge, whether about the locations of survivors or sensing capabilities of the searching agents, must be encoded in the form of an information map which specifies which areas agents should spend more time in when looking for survivors. In this work, we focus on the generation of mutual information-based maps for robots with binary detectors (i.e., that sense a 1 when a survivor is seen and a 0 otherwise), and demonstrate that, even with such limited sensing capabilities, the robots are able to hone in on the locations of multiple moving targets. We show that these information maps results in significantly lower mean absolute distance (MAD) than previously used maps through simulated search scenarios. Furthermore, we show that, using these maps, ergodic search can also outperform standard coverage-based methods for search.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

INTRODUCTION

Autonomous swarms of robots are becoming more prevalent in search and rescue operations due to robots' ability to explore areas unsafe for humans. However, search algorithms often struggle to fully utilize the collective capabilities of multiple agents or prior knowledge about locations of potential survivors. Methods which can take advantage of prior knowledge, like information-based methods, typically cannot scale to even moderately large swarms of robots [7]. In this work, we investigate ergodic search methods, which offer a good tradeoff between scalability and effectiveness.

Ergodic search is a recent method in the field of active information gathering. Active information gathering algorithms typically work by optimizing agents' trajectories in order to maximize an information metric like mutual information [7, 10, 16, 25]. These types of algorithms make full use of agents' sensing capabilities and prior knowledge, but have poor computational performance. Ergodic search, in contrast, treats the search task as a spatial coverage problem over an information map. Agents' trajectories are optimized so that the total amount of time spent by the agents in any region is proportional to the amount of information in that region, according to the map. While information metrics like mutual information typically require exponential time to compute [7], the ergodic metric is linear with respect to the number of agents.

While previous works have focused on ergodic control, we focus on how to create the information maps in the first place. Since the map defines the behavior of the multi-agent system, it should take into account both the agents' sensing capabilities and the current knowledge about the locations of survivors. To this end, we use the mutual information of a single sensor measurement to define our information map. By only considering a single measurement, we avoid the high computational complexity typically associated with mutual information, and instead rely on the ergodic metric to balance agents' exploration and exploitation. We derive equations to efficiently compute the maps for both single and multi-target search, where in the latter the number of targets is not known a priori. Though previous works have considered using a similar metric, expected Fisher information, to create maps for ergodic

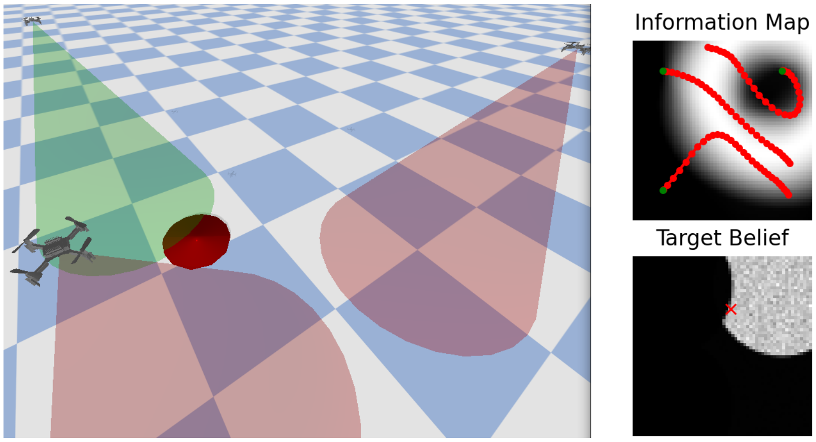

Figure 1.1: Snapshot of the simulation environment used to test and evaluate the proposed information map. The top left image shows the generated information map and ergodic trajectories for the three search agents (quadcopters). The bottom image shows the current belief about the target's (x, y) position.

search, this required strong assumptions about the measurement model and current distribution over potential survivor locations (e.g., differentiable and Gaussian). Our method only requires that the measurement model is known beforehand.

To show the effectiveness of these mutual information maps, we demonstrate that agents with only binary detectors can effectively localize multiple moving targets. The binary detectors sense a 1 when a survivor is detected and a 0 otherwise and thus provide little information about the survivors locations. This means that the agents must take full advantage of their sensing capabilities in order to localize the survivors. We show that our maps lead to much better accuracy (in terms of mean absolute distance) in the survivor's estimated locations compared to the maps used in prior ergodic search works. We also compare our method to a purely coverage-based method ("lawnmower" paths) and show that our method results in better accuracy.

A summary of the contributions in this work is as follows

1. A novel mutual information-based map for single target ergodic search with binary sensors.
2. Derivations of the cardinalized probability hypothesis density (CPHD) filtering equations for binary sensors to use for multi-target tracking.
3. A novel mutual information-based map for multi-target search based on the derived CPHD filter equations.

Chapter 2

RELATED WORK

2.1 Coverage Methods

We briefly mention coverage methods such as “lawn-mower” paths and frontier-based planners due to their well-established usage in search and other exploration problems. Lawn-mower trajectories, where agents move back and forth across the search space in a lawn-mower-like pattern, have been widely used for autonomous exploration to search for static targets like non-moving survivors or unexploded ordinances [3]. Though these paths are computationally cheap to generate and cover the entire search space, they do not generalize to the case where targets can move around. Also, full coverage becomes less useful when sensor noise, such as false detections or missed detections, are considered or some prior knowledge about the locations of survivors is known. Frontier methods generally suffer from the same issues, and so both are mainly limited to static search problems where sensor noise is low.

2.2 Active Information Gathering

The field of active information gathering (AIG) is focused on how information metrics, like mutual information and Fisher information, can be used to guide agents whose goal is to estimate some uncertain parameter (e.g., the locations of survivors). The basic premise of most of these metrics is to measure how much uncertainty will decrease after a sequence of actions are performed [25]. While mutual information measures this uncertainty in terms of entropy, Fisher information gives bounds on the variance instead.

Many different formulations and approximations for information have been considered in the AIG literature. In [16], the authors use a particle filter based estimate of α-mutual information (Rényi information) for fixed proximity sensors to create short time horizon plans for surveillance. Other works consider information maximizing control methods for mobile agents using various types of sensors [25, 11]. Due to computational constraints, these early works choose actions in a greedy manner, leading to myopic behavior that can result in failure to locate targets.

Recent works have developed less greedy methods using combination of computationally cheaper approximations of information gain and more efficient planning methods. In [7], the authors derive an approximate form for the mutual information when the measurement model is Gaussian, and alleviate some of the issues of greedy planning by carefully choosing the set of possible actions to avoid considering multiple trajectories which collect similar information. Other works consider sample-based planning techniques for information acquisition over long time horizons [12, 13]. Dames [10] considers many ways to make computing mutual information more computationally feasible—for example, the use of an adaptive cellular representation for the target belief distribution.

However, despite the these advancements, none of these AIG algorithms scale well to even moderately large multi agent systems without relying on decentralized planning. This is because the computational complexity of mutual information (along with the majority of other information measures) is exponential in the number of agents and the length of the planning horizon. The ergodic metric, in contrast, scales linearly with both the planning horizon and number of agents, making it better suited to multi-agent systems.

2.3 Ergodic Search

Ergodic search is another technique that has been applied to AIG problems. In the ergodic search framework, trajectories of agents are optimized so that the agents visit areas of the search space with a frequency proportional to the amount of information in that area [20]. An image depicting an ergodic trajectory is shown in Figure 2.1. This search strategy allows for the planning of long-term trajectories which balance between exploitation of areas of high information and exploration of low-information regions, even when the distribution of information is highly non-convex [24]. Typically, ergodic search works consider continuous action spaces [1, 2, 5, 6, 20, 19, 21, 22, 24], and therefore can better optimize trajectories compared to most other AIG methods which use coarsely discretized actions.

In [19] and [5], the authors use this framework for search with a heuristic-based map defined by the target belief distribution. In this work, we show that using the target belief distribution as the information map is not always effective as just maximizing the likelihood of observing the target may not be enough to get a good estimate of its location. Both [22] and [21] consider sensors with non-linear measurement models and use an information map defined with the Fisher information matrix determinant in order to get more informative measurements. Though maximizing the Fisher

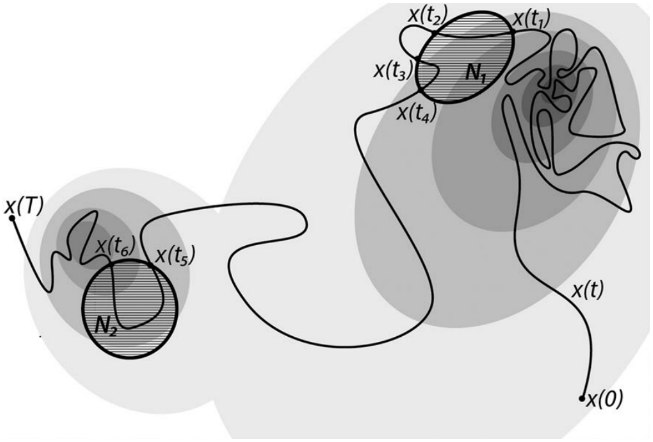

Figure 2.1: An example ergodic trajectory. The information map is shown as a contour plot in the background, and the trajectory as a black line. For each of the regions N_i, the amount of time spent in the region is proportional to the information in that region. Image taken from [24].

information is desirable, as it can be used to derive a lower bound for the variance of an unbiased estimator for a target location [9], it requires the measurement model to be differentiable. Otherwise, the Fisher Information does not exist at every point. For the binary sensors considered in this work, using Fisher information is particularly impractical as it is always either zero or undefined. We propose to use mutual information to generate these maps, which allows to more general sensors to be used.

Chapter 3

BACKGROUND

3.1 Ergodic Metric

In this section, we give an overview of the ergodic metric and how it can be optimized to create ergodic search trajectories. As the focus of this work is on generating information maps, there is no novelty presented in this section. Similar information can be found in [21, 24].

First, we define some notation. Let agent i's state at time t be $x_{i,t}$, and its location within the search space $y_{i,t}$. The set of all agent states is $\mathcal{X}$ and the set of all locations is $\mathcal{Y} \subset \mathbb{R}^n$. We assume the agents move according to the deterministic, discrete time dynamics $x_{i,t+1} = f(x_{i,t}, u_{i,t})$, where $u_{i,t}$ is a control input.

Now, assuming an information map $\Phi : \mathcal{Y} \to \mathbb{R}$ is given, we can plan a set of ergodic trajectories for the agents over a time horizon t_H. For brevity, we define $\mathbf{x} = [x_{0,t}, \dots, x_{N,t+t_H-1}]$ and $\mathbf{u} = [u_{0,t}, \dots, u_{N,t+t_H-1}]$. In ergodic search, the goal is to match the density of $\mathbf{x}$ with Φ. Formally, we want the following relationship:

$$\int_A \sum_{\tau=t}^{t+t_H-1} \sum_{i=1}^{N} \delta_{y_{i,\tau}}(y)dy = \int_A \Phi(y)dy \tag{3.1}$$

for any $A \subset \mathcal{Y}$, where $\delta_{y_{i,\tau}}$ is the dirac delta function centered at $y_{i,\tau}$. This says that the time spent by the agents in any region A of $\mathcal{Y}$, should be proportional to the total information in that region (according to Φ). In general, we assume Φ is normalized such that $\int_{\mathcal{Y}} \Phi(y)dy = 1$ and can define the spatial statistics of $\mathbf{x}$ as

$$C(y) = \frac{1}{Nt} \sum_{\tau=t}^{t+t_H-1} \sum_{i=1}^{N} \delta_{y_{i,\tau}}(y), \tag{3.2}$$

While it is unclear how we can optimize the controls $\mathbf{u}$ to get the relationship in (3.1) for an arbitrary set A, [20] derives a metric which is minimized when the relationship holds for spherical sets (i.e., the set A in (3.1) is a sphere in $\mathbb{R}^n$). First, let $\mathcal{F}_k$ be a Fourier transform over $\mathcal{Y}$. Note that, similar to [20], we use the cosine transform rather than the standard Fourier transform. Then, the ergodic metric is defined as

$$\mathcal{E}(\mathbf{x}) = \sum_{k \in (\mathbb{Z}^*)^n} \Lambda_k (\mathcal{F}_k(C) - \mathcal{F}_k(\Phi))^2, \tag{3.3}$$

where $\mathbb{Z}^*$ are the non-negative integers and

$$\Lambda_k = \frac{1}{(1 + ||k||^2)^{(n+1)/2}}, \tag{3.4}$$

where n is the dimension of the search space ($\mathcal{Y} \subset \mathbb{R}^n$). Minimizing the ergodic metric generally results in trajectories which approximate the relationship in (3.1), and have been shown to provide a good balance between exploration and exploitation [24].

As the goal of this work is to find useful information maps, we do not focus much on the optimization of $\mathcal{E}$ and instead rely on CasADi [4] as a black box optimization library. We do note that there are more efficient methods for optimizing the ergodic metric. For more information on these methods, see [23, 21].

3.2 Mutual Information

In this section, we give the definition and interpretation of mutual information as a measure for information gain. At a high level, the mutual information $I(V_1; V_2)$ between two random variables V_1, V_2 describes how much measuring one variable will decrease the uncertainty in the other. Mutual information is high when two variables are highly dependent on each other and zero when they are independent. It is symmetric in the sense that $I(V_1; V_2) = I(V_2; V_1)$. Mutual information can be defined in a few different, equivalent, ways:

$$\begin{aligned} I(V_1; V_2) &= \mathbb{E}_{p(V_2)}[D_{KL}(p(V_1|V_2) \; || \; p(V_1))] \\ &= D_{KL}(p(V_1, V_2) \; || \; p(V_1)p(V_2)) \\ &= H(V_1) - H(V_1|V_2), \end{aligned} \tag{3.5}$$

where $\mathbb{E}$ is the expected value, D_{KL} is the Kullback-Leibler (KL) divergence, and H is the Shannon entropy [9]. Line 2 of (3.5) relates to the interpretation as a measure of dependence between two the two variables, while line 3 gives the interpretation of a decrease in uncertainty (entropy).

One major advantage of mutual information compared to Fisher information is that no assumptions about the differentiability of the measurement model are required. Past methods have also relied on Gaussian assumptions for the relevant distributions in order to compute the Fisher information [24]. We show that the mutual information can be efficiently computed for arbitrary target belief distributions (e.g., a particle filter instead of a Kalman filter) with the binary sensors considered.

We also note that more general forms for mutual information exist, including α-mutual information, or Rényi information [15, 27]. However, unlike mutual information, there is no clear way to keep both interpretations (decrease in uncertainty and measure of dependence) at the same time [27], and so we only look at mutual information.

3.3 PHD Filters

Though on the surface multi-target tracking may appear to be a simple generalization of single target tracking, naïve implementations of multi-target filters have been shown to be intractable [14, 18, 26, 28]. As the number of targets grows, the size of the space containing their joint state grow exponentially. For example, if a single target's state could be represented as simply its $(x, y) \in \mathbb{R}^2$ coordinates on the plane, then the joint state of M targets would lie in $\left(\mathbb{R}^2\right)^M$.

For a particle filter representation, this implies that the number of particles required to effectively estimate the joint multi-target state scales exponentially with the number of targets [28]. Though Kalman filters seem promising, the number of parameters to be estimated would still scale quadratically with the number of targets. Additionally, the Kalman filter is not easily generalizable to the number of targets isn't known beforehand.

Probability hypothesis density (PHD) filters take a different approach to the problem to allow for the efficient estimation of a large and unknown number of target states. The basic premise behind PHD filters is to treat the joint multi-target density as an independent and identically distributed (IID) point process [17]. In an IID point process, the joint multi-target belief is approximated by a separable distribution:

$$b(\{s_1, s_2, \ldots, s_m\}) \approx (m!)c(m) \prod_{i=1}^{m} d(s_i). \tag{3.6}$$

Here, $b(\{s_1, s_2, \ldots, s_m\})$ is the estimated probability density (i.e., belief) for the set of target states s_1 through s_m. For a non-negative integer m, $c(m)$ is the probability that there are m targets and is called the "cardinality" distribution. $d(\cdot)$ is a probability distribution representing the density of targets. The m factorial is required in order to account for the possible permutations of states within the set.

Because of this approximation, the number of particles in a particle-based implementation of the PHD filter can scale linearly with the expected number of targets while still giving good estimates of the target density and number of targets [28]. In general, the PHD filtering equations allow for efficient multi-target density estimation.

We note that, usually, “the” PHD filter refers to the case where the cardinality distribution $c(\cdot)$ is Poisson since this leads to the simplest forms for the update equations. When $c(\cdot)$ is not Poisson, the filter is called a cardinalized PHD (CPHD) filter instead.

Chapter 4

SEARCH ENVIRONMENT

In this section, we define the search environment, including the agents, targets, and the binary sensors used by the agents to find and track the targets. The main purpose of this section is to define notation, and also define the measurement model of the binary sensors considered in this work.

4.1 Agents

We are to plan trajectories for N homogeneous agents. Denote agent i's state at time t by $x_{i,t} \in \mathcal{X}$ and its location within the search space $\mathcal{Y} \subset \mathbb{R}^n$ as $y_{i,t}$. We use $L_x : \mathcal{X} \to \mathcal{Y}$ to map from the agent's state to its location within the search space. We assume a deterministic discrete time dynamics model $x_{i,t+1} = f(x_{i,t}, u_{i,t})$, where $u_{i,t}$ is the control input at time t.

4.2 Targets

We consider homogeneous targets which move about the search space randomly. Call target j's state at time t $s^*_{j,t} \in \mathcal{S}$. We use $L_s : \mathcal{S} \to \mathcal{Y}$ to map from a target's state to its location within the search space. In the multi-target case, we will use M^* to denote the number of targets and $\mathbf{s}^*_t = \{s^*_{1,t}, \ldots, s^*_{M^*,t}\}$ the set of target states. Since it will be relevant later, we note that $\mathbf{s}^*_t$ is a member of the power set of $\mathcal{S}$: $P(\mathcal{S})$. For the single target case, we will drop the index j. We assume a known transition probability density $s^*_{j,t+1} \sim g(\cdot|s^*_{j,t})$ which describes how the target's state evolves over time. Note that while the agents know this probability density, they do not know the outcome of the random process.

4.3 Binary Sensors

In this work, we focus on binary sensors which measure a 1 when a target is detected and a 0 otherwise. We label the observation of agent i's sensor at time t as $z_{i,t}$. We call the sensor's field of view (FOV) $V(L_x(x)) \subseteq \mathcal{Y}$, which is the area of the search space $\mathcal{Y}$ visible when a robot is at location $L_x(x)$.

In the single target case, a sensor measures $z = 1$ with probability p_d when the target is within $V(L_x(x))$ (i.e., $L_s(s^*) \in V(L_x(x))$) and with probability p_f when the target is outside the sensor's FOV. p_d represents the probability of a true detection while p_f

is the probability of a false detection. To simplify notation, we will write $s^* \in V(x)$ instead of $L_s(s^*) \in V(L_x(x))$, noting that we are really talking about the robot's and target's location and not their full states. We call $h(z|s^*, x)$ the sensor model, which is a probability distribution over possible observations. For the single target setting, we have

$$h(z = 1|s^*, x) = \begin{cases} p_d & \text{if } s^* \in V(x) \\ p_f & \text{if } s^* \notin V(x) \end{cases} \tag{4.1}$$

$$h(z = 0|s^*, x) = \begin{cases} 1 - p_d & \text{if } s^* \in V(x) \\ 1 - p_f & \text{if } s^* \notin V(x) \end{cases}. \tag{4.2}$$

For the multi-target case, the sensor model becomes slightly more complex. We treat it such that a sensor has probability p_d of observing a single target within its FOV, and the probability of getting a reading of 1 goes up when there are more targets within the field of view. Let $\mathbf{1}_V(s)$ be the indicator function for $V(x)$ and

$$M_x^* = |V(x) \cap \mathbf{s}^*| = \sum_{j=1}^{M^*} \mathbf{1}_V(s_j^*) \tag{4.3}$$

the number of targets within the sensor's field of view from state x. Then we have the sensor model

$$h(z = 1|M_x^*, x) = \begin{cases} 1 - (1 - p_d)^{M_x^*} & \text{if } M_x^* > 0 \\ p_f & \text{if } M_x^* = 0 \end{cases} \tag{4.4}$$

$$h(z = 0|M_x^*, x) = \begin{cases} (1 - p_d)^{M_x^*} & \text{if } M_x^* > 0 \\ 1 - p_f & \text{if } M_x^* = 0 \end{cases}. \tag{4.5}$$

These models are used in filtering equations to estimate target locations as well as in the information maps derived in the following sections.

Chapter 5

SINGLE TARGET INFORMATION MAP

In this section, we derive the information map for single target search. The goal is to find a way of computing the mutual information of a single binary observation for a large number of positions in an efficient way. This is achieved by writing the mutual information in terms of the measure

$$\mu_V = \int_S \mathbf{1}_V(s) b_{t+1|t}(s) ds, \tag{5.1}$$

where $b_{t+1|t}(\cdot)$ is the probability density function representing the belief over target states, and $\mathbf{1}_V(\cdot)$ is the indicator function for the FOV $V(x)$ for an agent at state x. Rewriting in terms of μ_V will make computing the information map cheap (on the order of a couple milliseconds for a map of size 40 by 40).

5.1 The Information Map

Proposition 1. *The mutual information between the belief of target locations s and a single measurement z from an agent at state x is*

$$\begin{aligned} I_x(s;z) =& \mu_V \left(p_d \log\left(\frac{p_d}{p_1}\right) + (1-p_d)\log\left(\frac{1-T}{p_0}\right)\right) + \\ & (1-\mu_V)\left(p_f \log\left(\frac{p_f}{p_1}\right) + (1-p_f)\log\left(\frac{1-p_f}{p_0}\right)\right)' \end{aligned} \tag{5.2}$$

where $p_1 = p_d\mu_V + p_f(1-\mu_V)$ and $p_0 = 1 - p_1$ are the believed probabilities of observing a 1 *and* 0 *respectively.*

Proof. We start with the definition of mutual information using KL divergence in Eq. (3.5) and rewrite it to use the conditional distribution of z on s.

$$I_x(s;z) = D_{KL}(p(s,z;x) \mid\mid b(y)h(z;x)) \tag{5.3}$$

$$= \int_S \sum_{z\in\{0,1\}} p(s,z;x) \log\left(\frac{p(s,z;x)}{b(y)h(z;x)}\right) ds \tag{5.4}$$

$$\int_S \sum_{z\in\{0,1\}} h(z|s;x)b(s) \log\left(\frac{h(z|s;x)}{h(z;x)}\right) ds \tag{5.5}$$

We write $b(s)$ in place of $b_{t+1|t}$ for brevity. Then, we can write $h(z;x)$ in terms of μ_V, p_d, and p_d using the sensor model (equations (4.2) and (4.1)) and the law of total probability.

$$p_1 \triangleq h(1;x) = \int_S h(1|s;x)b(s)ds \tag{5.6}$$

$$= p_d \int_S \mathbf{1}_V b(s)ds + F \int_S (1 - \mathbf{1}_V)b(s)ds \tag{5.7}$$

$$= p_d \mu_V + p_f(1 - \mu_V) \tag{5.8}$$

Similarly,

$$p_0 \triangleq h(0;x) = 1 - h(1;x) \tag{5.9}$$

$$= 1 - p_d \mu_V - p_f(1 - \mu_V) \tag{5.10}$$

Then, (5.5) becomes

$$\int_S \left[h(0|s;x)b(s) \log\left(\frac{h(0|s;x)}{p_0}\right) + h(1|s;x)b(s) \log\left(\frac{h(1|s;x)}{p_1}\right)\right] ds \tag{5.11}$$

$$\begin{aligned} &= \int_S \mathbf{1}_V b(s) \left[p_d \log\left(\frac{p_d}{p_1}\right) + (1 - p_d) \log\left(\frac{1 - p_d}{p_0}\right)\right] ds + \\ &\int_S (1 - \mathbf{1}_V)b(s) \left[p_f \log\left(\frac{p_f}{p_1}\right) + (1 - p_f) \log\left(\frac{1 - p_f}{p_0}\right)\right] ds \end{aligned} \tag{5.12}$$

Since all the values in the square brackets are constant with respect to s, we can substitute in the definition for μ_V to get the final expression in (5.2):

$$\begin{aligned} I_x(s; z) =& \mu_V \left(p_d \log\left(\frac{p_d}{p_1}\right) + (1 - p_d) \log\left(\frac{1 - T}{p_0}\right)\right) + \\ &(1 - \mu_V) \left(p_f \log\left(\frac{p_f}{p_1}\right) + (1 - p_f) \log\left(\frac{1 - p_f}{p_0}\right)\right) \end{aligned}$$

□

5.2 Efficient Computation of μ_V

Now that we have the mutual information written in terms of μ_V, we discuss how this allows us to compute the mutual information efficiently. The first step is to discretize the belief over target locations onto a grid. In this work, $b_{t+1|t}(\cdot)$ is represented as a particle filter, so the simplest way is to just round the particle values to the nearest grid cell and sum up the weights of the particles in each cell. Next, we represent the indicator function for the FOV of an agent at state $x = 0$: $\mathbf{1}_0(s)$ on a grid as well, where the value of each cell with location y is 1 when $y \in V(0)$ and 0 when $y \notin V(0)$. Then we can rewrite μ_V as

$$\mu_V = \int_S \mathbf{1}_V(s) b_{t+1|t}(s) ds \tag{5.13}$$

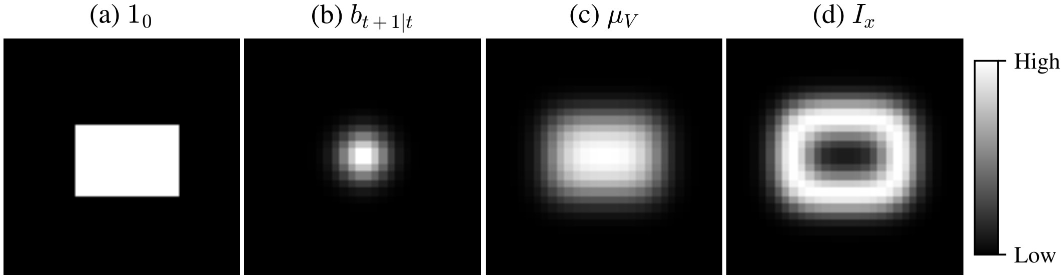

Figure 5.1: Process for computing the information map I_x. The mask $\mathbf{1}_0$ for a sensor with a rectangular FOV is shown in (a) and the current belief $b_{t+1|t}$ over target locations is shown in (b). μ_V is computed by convolving $\mathbf{1}_0$ and $b_{t+1|t}$. Then, equation (5.2) is used to compute the information map I_x.

$$= \int_S \mathbf{1}_0(s-x)b_{t+1|t}(s)ds \tag{5.14}$$

$$= (\mathbf{1}_0 * b_{t+1|t})(x), \tag{5.15}$$

where $*$ is the convolution operator. Because we can write μ_V in terms of the convolution operator and we can easily discretize both $b_{t+1|t}$ and $\mathbf{1}_0$, we can efficiently compute μ_V for every x on the grid using image convolution. Once we have compute μ_V, the information map can be computed using (5.2). This full process is summarized in Figure 5.1. Note that, somewhat counterintuitively, the information map is significantly different from the belief over the target's location (and is even low where the belief is high). This is because of the binary sensor model is highly non-linear with respect to location of a target.

Chapter 6

CPHD FILTER FOR BINARY SENSORS

In this section, we derive the correction equations (i.e., the update to the multi-target belief after making a measurement) for the CPHD filter from the sensor model described in section 4.3. We note that the predictor equations remain the same as those derived in [18, 17] with zero for the birth and death rates since we assume the number of targets stays the same.

6.1 CPHD Filter

In a CPHD filter, we estimate two distributions, one called the cardinality distribution c and the other the target density d. $c(m)$ is the believed probability of there being m targets in the search environment and $d(s)$ is a distribution over the states of the those targets. The assumption made in the CPHD filter is that the full belief over the set of the targets, which we will call $b(\mathbf{s})$, is an IID point process

$$b(\mathbf{s}) = (|\mathbf{s}|!)c(|\mathbf{s}|) \prod_{s \in \mathbf{s}} d(s). \tag{6.1}$$

We use an IID point process the approximate the true distribution over sets of targets because computing the full multi-target density is generally intractable [18, 28, 14, 26].

Using the common notation for Bayes filters, we write $c_{t+1|t}$ and $c_{t+1|t+1}$ for the estimate of the cardinality distribution before and after observation z_{t+1}. Similarly for the estimate of the target density we write $d_{t+1|t}$ and $d_{t+1|t+1}$.

As the predictor step $d_{t|t} \longrightarrow d_{t+1|t}$ remains unchanged from those shown in [18, 17], we refer the reader to that work for the details. This section focuses on the corrector step $d_{t+1|t} \longrightarrow d_{t+1|t+1}$, which are novel due the different type of sensor being considered compared to [17] and related works.

Correction Equations

At each time step t, we will want to update our belief from $b_{t+1|t}$ to $b_{t+1|t+1}$, which consists of updating $c_{t+1|t}$ to $c_{t+1|t+1}$ and $d_{t+1|t}$ to $d_{t+1|t+1}$.

The correction equation for the cardinality distribution is

$$c_{t+1|t+1}(m) = c_{t+1|t}(m) \times \begin{cases} (1-\mu_V p_d)^m - p_f(1-\mu_V)^m & \text{if } z_{t+1} = 0 \\ 1-(1-\mu_V p_d)^m + p_f(1-\mu_V)^m & \text{if } z_{t+1} = 1 \end{cases}, \tag{6.2}$$

where

$$\mu_V = \int_{\mathcal{S}} \mathbf{1}_V(s) d_{t+1|t}(s) ds \tag{6.3}$$

is the measure of of the FOV $V(x)$ under $d_{t+1|t}$ and $\mathbf{1}_V$ is the indicator function for $V(x)$.

The correction equation for the density distribution when $z_{t+1} = 0$ is

$$d_{t+1|t+1}(s) \propto d_{t+1|t}(s) \times \begin{cases} a_0 & \text{if } s \in V(x) \\ 1 & \text{if } s \notin V(x) \end{cases}, \tag{6.4}$$

where $a_0 = c_1/c_2$ and

$$c_1 = \sum_{m=1}^{\infty} c_{t+1|t}(m)(1-p_d)(1-p_d\mu_V)^{m-1} m \tag{6.5}$$

$$c_2 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left((1-p_d\mu_V)^{m-1} - p_f(1-\mu_V)^{m-1} \right) m, \tag{6.6}$$

Similarly, when $z_{t+1} = 1$, the update is

$$d_{t+1|t+1}(s) \propto d_{t+1|t}(s) \times \begin{cases} a_1 & \text{if } s \in V(x) \\ 1 & \text{if } s \notin V(x) \end{cases}, \tag{6.7}$$

where $a_1 = c_3/c_4$

$$c_3 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left(1 - (1-p_d)(1-p_d\mu_V)^{m-1} \right) m \tag{6.8}$$

$$c_4 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left(1 - (1-p_d\mu_V)^{m-1} + p_f(1-\mu_V)^{m-1} \right) m \tag{6.9}$$

Derivations

Best IID Point Process Approximation

In order to derive the CPHD correction equations, we must first generalize theorem 4 (Best Poisson Approximation) from [18] to the case where the cardinality distribution $c(m)$ is arbitrary. First, we define some terminology. Let $\mathbf{s} = \{s_1, \ldots, s_{|\mathbf{s}|}\} \subset \mathcal{S}$

be the random finite set describing the joint multi-target state, and $f(\mathbf{s})$ a probability distribution over $P(\mathcal{S})$ (the power set of $\mathcal{S}$). $D_f(s)$ is called the probability hypothesis density (PHD) for f and is uniquely defined by the equation [18]

$$\int_A D_f(s)ds = \mathbb{E}_f\left[|A \cap \mathbf{s}|\right], \tag{6.10}$$

for all $A \subset \mathcal{S}$. $D_f(s)$ is the expected density of targets whose integral over the target state space $\mathcal{S}$ gives the expected total number of targets.

Now we consider theorem 4 from [18], which states that $D_f(s)$ is the best (in the information theoretic sense) Poisson approximation to $f(\mathbf{s})$. Formally, it is shown that the Kullback-Leibler (KL) divergence

$$D_{KL}\left(f(\mathbf{s}) \,\middle\|\, (m!)\frac{\lambda^m e^{-\lambda}}{m!}\prod_{i=1}^{m} d(s_i)\right) \tag{6.11}$$

is minimized when $d(s) = (1/\lambda)D_f(s)$ and $\lambda = \mathbb{E}_f[m]$. In this paper, we consider the more general case where the distribution of the number of targets $c(\cdot)$ is an arbitrary distribution (not necessarily Poisson) and want to show that the KL divergence is still minimized when $d(s) \propto D_f(s)$.

Proposition 2 (Best IID Point Process Approximation)**.** *We show that*

$$D_{KL}\left(f(\mathbf{s}) \,\middle\|\, (m!)c(m)\prod_{i=1}^{m} d(s_i)\right) \tag{6.12}$$

is minimized when

$$c(m) = f(|\mathbf{s}| = m) \triangleq \frac{1}{m!}\int_{\mathcal{S}^m} f(\{s_1, \dots, s_m\})ds_1 \dots ds_m \tag{6.13}$$

and $d(s) \propto D(s)$.

Proof. The proof is similar to the proof of theorem 4 in [18]:

$$D_{KL}\left(f(\mathbf{s}) \,\middle\|\, (m)c(m)\prod_{i=1}^{m} d(s_i)\right) \tag{6.14}$$

$$= \int f(\mathbf{s}) \log\left(\frac{f(\mathbf{s})}{(m!)c(m)\prod_{i=1}^{m} d(s_i)}\right)\delta\mathbf{s} \tag{6.15}$$

$$= a - \int f(\mathbf{s})\left(\log(c(m)) + \sum_{i=1}^{m}\log(d(s_i))\right)\delta\mathbf{s}, \tag{6.16}$$

where a is constant with respect to $c(\cdot)$ and $d(\cdot)$. We can now look at the contributions from $c(\cdot)$ and $d(\cdot)$ separately. First, we have

$$-\int f(\mathbf{s})\log(c(m))\delta\mathbf{s} \tag{6.17}$$

$$= -\sum_{m=0}^{\infty}\frac{1}{m!}\int_{S^M} f(\{s_1,\dots,s_m\})\log(c(m))ds_1\dots ds_m \tag{6.18}$$

by the definition of a set integral. Therefore we have

$$= -\sum_{m=0}^{\infty} f(|\mathbf{s}| = m)\log(c(m)). \tag{6.19}$$

Minimizing this equation is equivalent to minimizing $D_{KL}(f(|\mathbf{s}| = m) \mid\mid c(m))$, so the optimal value for $c(m)$ is $f(|\mathbf{s}| = m)$.

Now, looking at the contribution from $d(\cdot)$, we have

$$-\int f(\mathbf{s})\sum_{i=1}^{M}\log(d(s_i))\delta\mathbf{s} = -\int_S D_f(s)\log(d(s))ds \tag{6.20}$$

by proposition 2a from [18]. Since $D_f(s)$ must be non-negative, we can write it as $\lambda d^*(s)$, where $\lambda = \int_S D_f(s)ds > 0$ and $d^*(s)$ is a probability density function. So, we have

$$= -\lambda\int_S d^*(s)\log(d(s))ds. \tag{6.21}$$

Minimizing this is equivalent to minimizing $D_{KL}(d^*(s) \mid\mid d(s))$, so we have a minimum when

$$d(s) = d^*(s) \propto D_f(s). \tag{6.22}$$

□

Cardinality Distribution Corrector

Using proposition 2, the update to the cardinality distribution $c_{t+1|t+1}(m)$ should be

$$c_{t+1|t+1}(m) = b\left(|\mathbf{s}| = m|z_{t+1},\dots,z_1\right) \tag{6.23}$$

We are leaving out the explicit dependence of the belief on x (i.e., we are writing $b(\mathbf{s}|z)$ instead of $b(\mathbf{s}|z,x)$) to reduce clutter. Next,

$$= \frac{1}{m!}\int_{S^m} b(\mathbf{s}_{m,t+1}|z_{t+1},\dots,z_1)d\mathbf{s}_{m,t+1}, \tag{6.24}$$

where $\mathbf{s}_{m,t+1} \triangleq \{s_{1,t+1}, \ldots, s_{m,t+1}\}$ and $d\mathbf{s}_{m,t+1} \triangleq ds_{1,t+1} \ldots ds_{m,t+1}$ are, again, used to reduce clutter.

$$\propto \frac{1}{m!} \int_{\mathcal{S}^m} h(z_{t+1}|\mathbf{s}_{m,t+1}) b(\mathbf{s}_{m,t+1}|z_t, \ldots, z_1) d\mathbf{s}_{m,t+1} \tag{6.25}$$

$$\approx \frac{1}{m!} \int_{\mathcal{S}^m} h(z_{t+1}|\mathbf{s}_{m,t+1}) b_{t+1|t}(\mathbf{s}_{m,t+1}) d\mathbf{s}_{m,t+1}. \tag{6.26}$$

Recognizing that the sensor model $h(z_{t+1}|\mathbf{s}_{m,t+1})$ only depends on the number of targets within the FOV $V(x)$,

$$M_x \triangleq |V(x) \cap \mathbf{s}_{m,t+1}|, \tag{6.27}$$

we can rewrite this as

$$= \frac{1}{m!} \sum_{m=0}^{m} h(z_{t+1}|M_x(\mathbf{s})) \int_{\mathcal{S}^m, M_x = m} b_{t+1|t}(\mathbf{s}_{m,t+1}) d\mathbf{s}_{m,t+1}. \tag{6.28}$$

Since $b_{t+1|t}(\mathbf{s}_{m,t+1})$ is a distribution describing an IID point process, $M_x \sim \text{Binomial}(m, \mu_V)$, and so we have

$$= c_{t+1|t}(m) \sum_{i=0}^{m} h(z_{t+1}|M_x = i) \binom{m}{i} (\mu_V)^i (1 - \mu_V)^{m-i} \tag{6.29}$$

For $z_{t+1} = 0$, this is

$$= c_{t+1|t}(m) \left[(1 - p_f)(1 - \mu_V)^m + \sum_{i=1}^{m} (1 - p_d)^i \binom{m}{i} (\mu_V)^i (1 - \mu_V)^{m-i} \right]. \tag{6.30}$$

Using the binomial theorem we simplify this to

$$= c_{t+1|t}(m) \left[(1 - p_f)(1 - \mu_V)^m + (1 - \mu_V p_d)^m - (1 - \mu_V)^m \right] \tag{6.31}$$

$$= c_{t+1|t}(m) \left[(1 - \mu_V p_d)^m - p_f (1 - \mu_V)^m \right] \tag{6.32}$$

The case for $z_{t+1} = 1$ follows similarly and gives us

$$= c_{t+1|t}(m) \left[1 - (1 - \mu_V p_d)^m + p_f (1 - \mu_V)^m \right] \tag{6.33}$$

Target Density Corrector

For the target density corrector, proposition 2 showed that $d_{t+1|t+1}(s)$ should be proportional to $D_{b(\mathbf{s}|z_{t+1}, \ldots, z_1)}(s)$ in order to get the best IID point process approximation for $b(\mathbf{s}_{t+1}|z_{t+1}, \ldots, z_1)$. First, for all $A \subset \mathcal{S}$, we have

$$\int_A D_{b(\mathbf{s}_{t+1}|z_{t+1}, \ldots, z_1)}(s) = \mathbb{E}_{b(\cdot|z_{t+1}, \ldots, z_1)} \left[|A \cap \mathbf{s}_{t+1}| \right] \tag{6.34}$$

To reduce clutter, we will drop the time index on $\mathbf{s}_{t+1}$, making it just $\mathbf{s}$. Then, we have

$$= \int b(\mathbf{s}|z_{t+1}, \ldots, z_1)|A \cap \mathbf{s}|\delta\mathbf{s} \tag{6.35}$$

$$\propto \int h(z_{t+1}|\mathbf{s}, z_t, \ldots, z_1) b(\mathbf{s}|z_t, \ldots, z_1)|A \cap \mathbf{s}|\delta\mathbf{s} \tag{6.36}$$

$$\approx \int h(z_{t+1}|\mathbf{s}, z_t, \ldots, z_1) b_{t+1|t}(\mathbf{s})|A \cap \mathbf{s}|\delta\mathbf{s} \tag{6.37}$$

$$= \int h(z_{t+1}|\mathbf{s}) b_{t+1|t}(\mathbf{s})|A \cap \mathbf{s}|\delta\mathbf{s}. \tag{6.38}$$

We can rewrite $|A \cap \mathbf{s}|$ as

$$\sum_{i=1}^{|\mathbf{s}|} \mathbf{1}_A(s_j), \tag{6.39}$$

and use the fact that $b_{t+1|t}(\mathbf{s})$ represents an IID point process, to give us

$$= \sum_{m=1}^{\infty} c_{t+1|t}(m) \sum_{i=1}^{m} \int_{\mathcal{S}^m} h(z_{t+1}|s_1, \ldots, s_m) \mathbf{1}_A(s_i) \left(\prod_{j=1}^{m} d_{t+1|t}(s_j) \right) ds_1 \ldots ds_m. \tag{6.40}$$

For $z_{t+1} = 0$,

$$h(0|M_x, x) = \begin{cases} (1 - p_d)^{M_x} & \text{if } M_x > 0 \\ 1 - p_f & \text{if } M_x = 0 \end{cases}, \tag{6.41}$$

where

$$M_x = |V(x) \cap \mathbf{s}| = \sum_{j=1}^{m} \mathbf{1}_V(s_j) \tag{6.42}$$

is the number of targets in $\mathbf{s}$ within the FOV $V(s)$ and $\mathbf{1}_V$ is the indicator function for $V(x)$. Now, we can rewrite the integral in (6.40) as

$$\int_{\mathcal{S}^m} h(z_{t+1}|s_1, \ldots, s_m) \mathbf{1}_A(s_i) \left(\prod_{j=1}^{m} d_{t+1|t}(s_j) \right) ds_1 \ldots ds_m \tag{6.43}$$

$$\begin{aligned} &= \int_{\mathcal{S}^m, M_x > 0} (1 - p_d)^{M_x} \mathbf{1}_A(s_i) \left(\prod_{j=1}^{m} d_{t+1|t}(s_j) \right) ds_1 \ldots ds_m \\ &\quad + \int_{\mathcal{S}^m, M_x = 0} (1 - p_f) \mathbf{1}_A(s_i) \left(\prod_{j=1}^{m} d_{t+1|t}(s_j) \right) ds_1 \ldots ds_m \end{aligned} \tag{6.44}$$

Using the definition of M_x we can rewrite this as

$$\begin{aligned}=&\int_{\mathcal{S}^m, M_x>0} \mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-p_d)^{\mathbf{1}_V(s_j)} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m \\ &+\int_{\mathcal{S}^m, M_x=0} (1-p_f)\mathbf{1}_A(s_i)\left(\prod_{j=1}^{m} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m\end{aligned} \tag{6.45}$$

$$\begin{aligned}=&\int_{\mathcal{S}^m} \mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-p_d)^{\mathbf{1}_V(s_j)} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m \\ &-\int_{\mathcal{S}^m, M_x=0} \mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-p_d)^{\mathbf{1}_V(s_j)} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m \\ &+\int_{\mathcal{S}^m, M_x=0} (1-p_f)\mathbf{1}_A(s_i)\left(\prod_{j=1}^{m} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m.\end{aligned} \tag{6.46}$$

$$\begin{aligned}=&\int_{\mathcal{S}^m} \mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-p_d)^{\mathbf{1}_V(s_j)} d_{t+1|t}(s_j)\right) ds_1 \dots ds_m \\ &-\int_{\mathcal{S}^m} \mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-\mathbf{1}_V(s_j)) d_{t+1|t}(s_j)\right) ds_1 \dots ds_m \\ &+\int_{\mathcal{S}^m} (1-p_f)\mathbf{1}_A(s_i)\left(\prod_{j=1}^{m}(1-\mathbf{1}_V(s_j)) d_{t+1|t}(s_j)\right) ds_1 \dots ds_m.\end{aligned} \tag{6.47}$$

Now, we write this equation in terms of the measures $\mu_{A\cap V}$ and $\mu_{A\setminus V}$ over $d_{t+1|t}$, which follow the same definition as (6.3):

$$\begin{aligned}=&(\mu_{A\cap V}(1-p_d)+\mu_{A\setminus V})((1-p_d)\mu_V+(1-\mu_V))^{m-1} \\ &-(\mu_{A\setminus V})(1-\mu_V)^{m-1}+(1-p_f)(\mu_{A\setminus V})(1-\mu_V)^{m-1}\end{aligned} \tag{6.48}$$

$$= (\mu_{A\cap V}(1-p_d)+\mu_{A\setminus V})(1-p_d\mu_V)^{m-1} - p_f(\mu_{A\setminus V})(1-\mu_V)^{m-1} \tag{6.49}$$

Rearranging into terms related to $\mu_{A\cap V}$ and $\mu_{A\setminus V}$, we get

$$= \mu_{A\cap V}(1-p_d)(1-p_d\mu_V)^{m-1} + \mu_{A\setminus V}\left((1-p_d\mu_V)^{m-1} - p_f(1-\mu_V)^{m-1}\right) \tag{6.50}$$

Now, plugging the whole thing into the sum from (6.40), we get

$$= \mu_{A\cap V}c_1 + \mu_{A\setminus V}c_2, \tag{6.51}$$

where

$$c_1 = \sum_{m=1}^{\infty} c_{t+1|t}(m)(1 - p_d)(1 - p_d \mu_V)^{m-1} m \tag{6.52}$$

$$c_2 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left((1 - p_d \mu_V)^{m-1} - p_f (1 - \mu_V)^{m-1} \right) m, \tag{6.53}$$

are constants with respect to A. Since this should be true for all $A \subset \mathcal{S}$, the update to the density $d_{t+1|t}$ is

$$d_{t+1|t+1}(s) \propto d_{t+1|t}(s) \times \begin{cases} c_1 & \text{if } s \in V(x) \\ c_2 & \text{if } s \notin V(x) \end{cases}. \tag{6.54}$$

Since this requires normalization at the end, we can equivalently write the update as

$$d_{t+1|t+1}(s) \propto d_{t+1|t}(s) \times \begin{cases} c_1/c_2 & \text{if } s \in V(x) \\ 1 & \text{if } s \notin V(x) \end{cases}. \tag{6.55}$$

For the case where $z_{k+1} = 1$, the derivations are similar and we get

$$d_{t+1|t+1}(s) \propto d_{t+1|t}(s) \times \begin{cases} c_3/c_4 & \text{if } s \in V(x) \\ 1 & \text{if } s \notin V(x) \end{cases}, \tag{6.56}$$

where

$$c_3 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left(1 - (1 - p_d)(1 - p_d \mu_V)^{m-1} \right) m \tag{6.57}$$

$$c_4 = \sum_{m=1}^{\infty} c_{t+1|t}(m) \left(1 - (1 - p_d \mu_V)^{m-1} + p_f (1 - \mu_V)^{m-1} \right) m \tag{6.58}$$

In general, we won't be able to simplify these equations further, so they are the final equations for the CPHD filter. In practice, this computation is cheap for a distribution with a reasonable maximum number of targets such that $c_{t+1|t}(m) = 0$ for all m greater than a certain number.

Chapter 7

MULTI-TARGET INFORMATION MAP

In this section, we derive the information map used for multi-target search. The idea here is to find the mutual information between the target density $d(s)$ and the next observation z_{t+1} made by an agent. To reduce clutter, we will write z in place of z_{t+1}. Similar to the single target mutual information map (chapter 5), we compute the information for every point on a grid in order to take advantage of the fast computation of convolution operations.

Proposition 3.

$$I_x(s;z) = \sum_{z\in\{0,1\}} h(z)\left[\frac{a_z\mu_V}{1+(a_z-1)\mu_V}\log(a_z) - \log(1+(a_z-1)\mu_V)\right], \tag{7.1}$$

where (same as (6.3)*)*

$$\mu_V = \int_S \mathbf{1}_V(s) d_{t+1|t}(s) ds$$

and (from section 6.1)

$$a_0 = \frac{\sum_{m=1}^{\infty} c_{t+1|t}(m)(1-p_d)(1-p_d\mu(x))^{m-1}m}{\sum_{m=1}^{\infty} c_{t+1|t}(m)\left((1-p_d\mu(x))^{m-1} - p_f(1-\mu(x))^{m-1}\right)m} \tag{7.2}$$

$$a_1 = \frac{\sum_{m=1}^{\infty} c_{t+1|t}(m)\left(1-(1-p_d)(1-p_d\mu(x))^{m-1}\right)m}{\sum_{m=1}^{\infty} c_{t+1|t}(m)\left(1-(1-p_d\mu(x))^{m-1} + p_f(1-\mu(x))^{m-1}\right)m}. \tag{7.3}$$

$h(z)$ is estimated using the current of target density and cardinality distribution $d_{t+1|t}$ and $c_{t+1|t}$ and is given as

$$h(0) = \sum_{m=0}^{\infty} c_{t+1|t}\left[(1-p_d\mu(x))^m - p_f(1-\mu(x))^m\right] \tag{7.4}$$

$$h(1) = \sum_{m=0}^{\infty} c_{t+1|t}\left[1-(1-p_d\mu(x))^m + p_f(1-\mu(x))^m\right] \tag{7.5}$$

Though this equation does not simplify down as nicely as in the single target case, it is still efficient to compute, taking on average a few milliseconds at each time step in our simulations.

Proof. This information map is derived from the update equations written in section section 6.1. This approximates the true mutual information

$$\sum_{z\in\{0,1\}} \int_{\mathcal{S}} d(s|z)h(z)\log\frac{d(s|z)}{d(s)} \tag{7.6}$$

by using the update from $d_{t+1|t}(s)$ to $d_{t+1|t+1}$ in place of $d(s|z)$. This gives us

$$I_x(s;z_{t+1}) \triangleq \sum_{z_{t+1}\in\{0,1\}} h(z_{t+1}) \int_{\mathcal{S}} d_{k+1|k+1}(s)\log\frac{d_{t+1|t+1}(s)}{d_{t+1|t}(s)} \tag{7.7}$$

$$\begin{aligned} = \sum_{z_{t+1}\in\{0,1\}} h(z_{t+1}) \Bigg[\int_{\mathcal{S}\cap V(x)} & d_{k+1|k+1}(s)\log\frac{d_{t+1|t+1}(s)}{d_{t+1|t}(s)} + \\ \int_{\mathcal{S}\setminus V(x)} & d_{k+1|k+1}(s)\log\frac{d_{t+1|t+1}(s)}{d_{t+1|t}(s)} \Bigg] \end{aligned} \tag{7.8}$$

Now, normalizing the update equations from section 6.1, we can see that

$$d_{k+1|k+1}(s) = \begin{cases} \frac{a_0}{1+(a_0-1)\mu_V} & \text{when } z_{t+1}=0 \\ \frac{a_1}{1+(a_1-1)\mu_V} & \text{when } z_{t+1}=1 \end{cases}, \tag{7.9}$$

so (7.8) becomes

$$\begin{aligned} = \sum_{z_{t+1}\in\{0,1\}} h(z_{t+1}) \Bigg[& \frac{a_z\mu_V}{1+(a_z-1)\mu_V}\log(\frac{a_z}{1+(a_z-1)\mu_V}) + \\ & \frac{1-\mu_V}{1+(a_z-1)\mu_V}\log\frac{1}{1+(a_z-1)\mu_V)} \Bigg] \end{aligned} \tag{7.10}$$

$$= \sum_{z_{t+1}\in\{0,1\}} h(z_{t+1}) \left[\frac{a_z\mu_V}{1+(a_z-1)\mu_V}\log(a_z) - \log(1+(a_z-1)\mu_V) \right] \tag{7.11}$$

□

Chapter 8

EXPERIMENTS

8.1 Experimental Setup

In order to compare ergodic search with our mutual information maps, we simulate search scenarios with varying numbers of targets and agents. In both the single and multi-target experiments, the agents and targets are enclosed in a 6 by 6 meter space. The targets are represented as red cylinders, though in real life these would be human survivors or other targets of interest. The agents are quadcopters which fly at a fixed height above the ground and each have one binary sensor. We assume that position control is reasonably good for the low velocities the drones will move at, and so the planned trajectories for the agents are simply a series of waypoints. A picture of the setup with three targets and two agents is shown in Figure 8.1.

In the simulations, the targets are either static or moves according to a random walk. The random walk is such that the velocity of the target in the x and y directions change by a small random value drawn from a normal distribution with standard deviation 0.0001 ten times a second. The maximum magnitude of the velocity is clipped at 0.015 m/s, and target which hit the boundary of the environment have their velocity reset to zero.

The agents' sensors take measurements at 10 Hz. As described in section 4.3, the agent's binary sensors are noisy in the sense that they can get both false detections and missed detections. When a single target is within the FOV of an agent, its

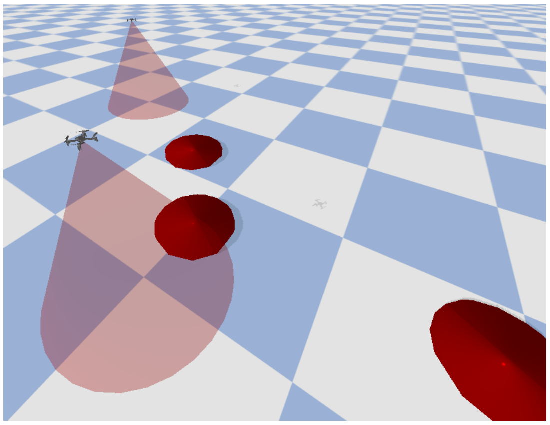

Figure 8.1: Experimental setup with three targets and two agents

sensor has probability $p_d = 0.8$ of getting a true detection and 0.2 of not detecting the target. When not observing any targets, they have a probability $p_f = 0.01$ of getting a false detection and 0.99 of getting a true negative. The sensor model when multiple targets are in the FOV of the agent can be found in section 4.3, with the same values for p_d and p_f mentioned above.

In the single target experiments, the target is tracked with a simple particle filter using 60,000 particles. Information on particle filter implementations can be found in, for example, [8]. This number of particles is overkill, but contributes relatively little to computation cost. In the multi-target experiments, the targets are tracked with a particle implementation of the CPHD filter described in section 6.1, again with 60,000 particles.

We compare three different methods:

1. **Ergodic MI**: Ergodic search with our mutual information map (either the single target or multi-target version depending on the experiment). The information map is computed over a 40 by 40 grid, and 10×10 Fourier components are used to optimize the ergodic trajectories. At each time step, sensor measurements are used to update the particle filter. Then, the information map is updated and ergodic trajectories replanned.

2. **Ergodic Density**: Ergodic search with the target density used as the information map. This does not take into account the sensor model when performing the search. The same specifications as method **Ergodic MI** are used for the ergodic search procedure.

3. **Lawnmower**: A lawn-mower trajectory. For multiple agents, the path is just split evenly between the agents so their paths do not overlap.

8.2 Single Target Results

First, we compare the three methods in simulations with a single moving target. The starting location of the target is randomized according to a uniform distribution over the search space. We use mean absolute distance (MAD) to measure how accurate the belief over target states is. For a particle filter with particles s_i and weights w_i, the MAD is computed as

$$\sum_i w_i ||s_i - s^*||_2, \tag{8.1}$$

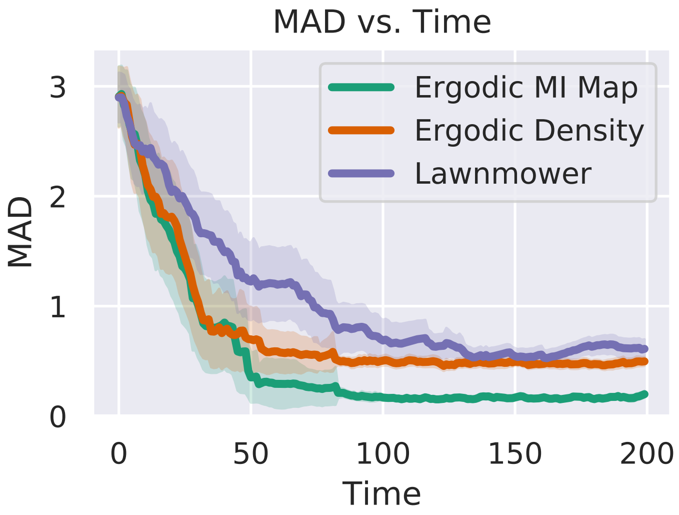

Figure 8.2: Comparison of different methods for tracking a single moving target with two agents in terms of MAD. Highlighted areas show ± 2 standard errors (standard deviation of the mean).

where s^* is the true target location and $||s_i - s^*||_2$ is the Euclidean distance. Figure 8.2 shows the results from 40 simulations of each method over 200 time steps (20 seconds) with two agents. As can be seen, **Ergodic MI** results in much lower steady state error than the others as the agents are able to better utilize their sensors. The **Lawnmower** method, as expected, performs the worst since the targets move over time.

Next, we look at the effect of the number of agents on the quality of the search while using method **Ergodic MI**. The results from 40 simulations of each scenario are plotted in Figure 8.3. As can be seen, there are diminishing returns for larger teams of agents. While the difference between a single searching agent and two searching agents is quiet large, more agents seems to provide relatively little decrease in the MAD over time.

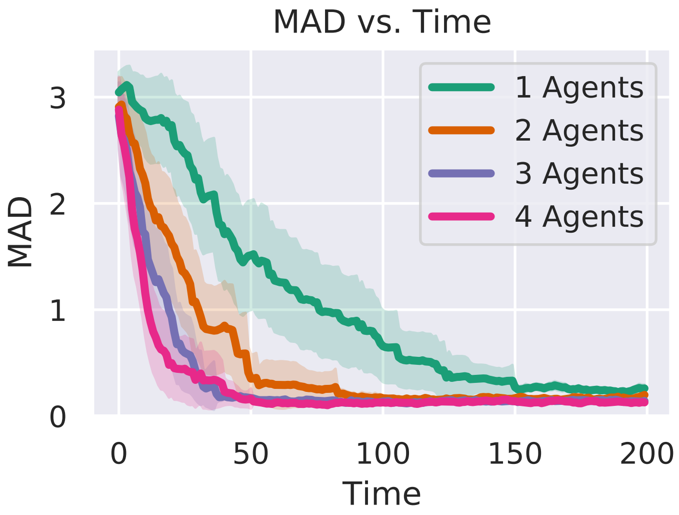

Figure 8.3: MAD vs. time for varying numbers of agents and a single target using method **Ergodic MI**. Highlighted areas show ± 2 standard errors (standard deviation of the mean).

8.3 Multi-Target Results

For the multi-target scenarios, we need to measure MAD in a different way. We use the particle filter density estimate from the CPHD filter to do this. For each particle, we find which target it is closest to and have it only contribute to the MAD for that target. Then, we average the individual MADs to give the reported MAD. Figure 8.4 visualizes how the particles are separated by target.

Now, we first compare the MAD over time for the three listed methods. The cardinality distribution $c(m)$ is initialized so that there is an equal probability of being between 0 and 9 targets. The expected number of targets is just a mean over this distribution, i.e.,

$$\sum_{m=0}^{9} mc(m). \tag{8.2}$$

In this comparison, the true number of targets is three and the targets initial locations are randomized uniformly over the search space. The results of 40 simulations of each method are shown in Figure 8.5. As can be seen, the **Ergodic MI** method performs the best both in terms of MAD and the expected number of targets. Both the **lawnmower** method and **ergodic density** method fail to keep track of the targets locations and therefore also have worse estimates of the number of targets.

Next, we look at how the performance is affected by the number of searching agents.

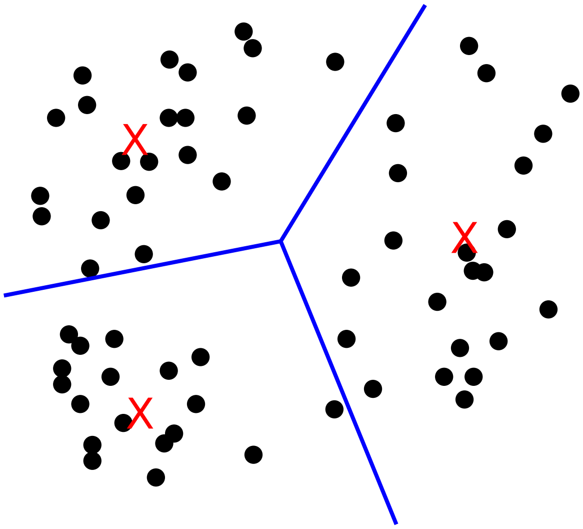

Figure 8.4: How MAD is computed in the multi target setting. Each • is a particle and each X a target. The MAD is computed for each target individually based on the particles that are closest to it (blue lines show separation), and then all the values are averaged to give the final MAD.

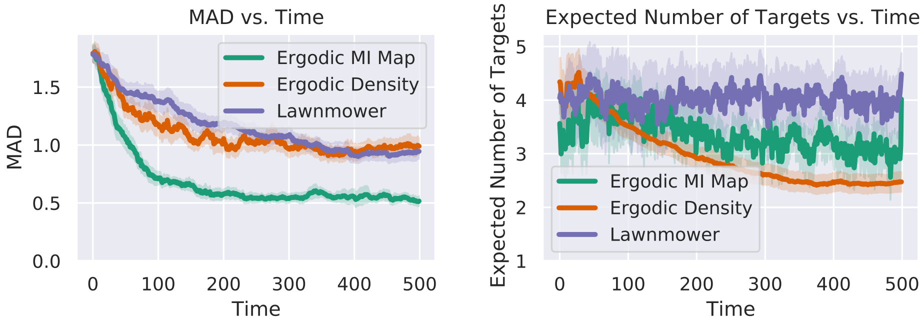

Figure 8.5: Comparisons between three methods with three targets.

Figure 8.6 shows this comparison for 40 simulations with three targets in each one. Although the MAD follows the same trend seen the single target search simulations, with more agents resulting in better performance, the expected number of targets over time gets worse with the number of agents. This is a limitation of our derived CPHD filtering method. As noted in [18], the PHD filter does a poor job of estimating the number of targets when the sensors have low signal to noise. Though the CPHD is proposed to solve this issue [17], it does not guarantee that the expected number of target over time will be unbiased. In our experiments, we found that the CPHD filter tended to overestimate the number of targets in most situations.

Finally, we look at how the performance of two searching agents varies with the

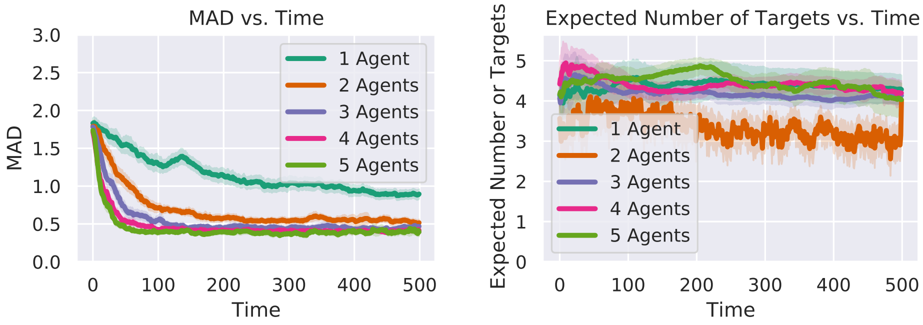

Figure 8.6: Comparisons for different numbers of searching agents with three targets.

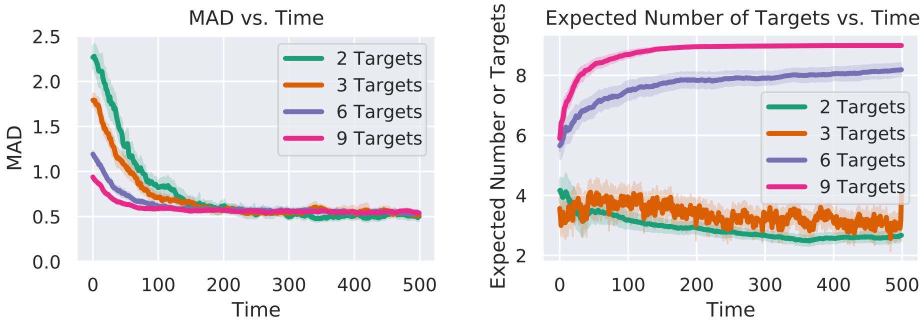

Figure 8.7: Comparisons for different numbers of targets with two searching agents.

number of targets. The result from 40 simulations with two agents and 2, 3, 6, and 9 targets are shown in Figure 8.7. As can be expected, the expected number of targets is more difficult to estimate with more targets per agent. Interestingly, the MAD seems to converge to around the same value regardless of the number of targets, though this is somewhat misleading because the MAD naturally is higher for fewer number of targets. Figure 8.8 shows a "summed" version of MAD, where instead of averaging the individual target MADs, they are instead added together. This runs into the opposite issue of the averaging version, making it so that more targets results in a naturally higher value. More important is the decrease in the MAD over time, which, as expected, is greater for fewer targets.

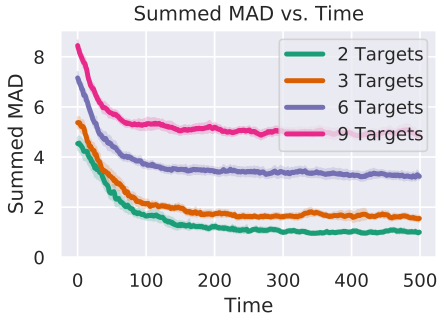

Figure 8.8: Comparisons of “summed” MAD for different numbers of targets with two searching agents.

Chapter 9

CONCLUSIONS

In this work, we derived mutual information maps for both single and multi-target search with binary sensors. Additionally, we derived the CPHD filtering equations for these sensors in order to track an unknown number of targets. We show that our maps results in better target localization accuracy than a naive map which does not consider the agent's sensor model. We also demonstrate that a "lawnmower" coverage method has significantly worse performance.

This work shows that using information-theoretic metrics to create maps for ergodic search can significantly increase the search performance. This is especially important in search and rescue operations, where it is necessary to quickly and accurately locate survivors.

In future works, we would like to consider other sensing modalities to allow our method to be used in wider variety of situations. In addition, we would like to improve the multi-target filtering equations to get better estimates of the number of targets as this appears to be the biggest weakness of our method.

BIBLIOGRAPHY

[1] I. Abraham et al. "Ergodic Exploration Using Binary Sensing for Nonparametric Shape Estimation". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 827–834.

[2] Ian Abraham, Anastasia Mavrommati, and Todd Murphey. "Data-Driven Measurement Models for Active Localization in Sparse Environments". In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania, June 2018. DOI: 10.15607/RSS.2018.XIV.045.

[3] Ercan U Acar et al. "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods". In: *The International journal of robotics research* 22.7-8 (2003), pp. 441–466.

[4] Joel A E Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.

[5] Elif Ayvali, Hadi Salman, and Howie Choset. "Ergodic coverage in constrained environments using stochastic trajectory optimization". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 5204–5210.

[6] Elif Ayvali et al. "Utility-guided palpation for locating tissue abnormalities". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 864–871.

[7] Benjamin Charrow, Vijay Kumar, and Nathan Michael. "Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information". In: *Robotics: Science and Systems IX* (2013). DOI: 10.15607/rss.2013.ix.053.

[8] Zhe Chen et al. "Bayesian filtering: From Kalman filters to particle filters, and beyond". In: *Statistics* 182.1 (2003), pp. 1–69.

[9] T. M. Cover and Joy A. Thomas. *Elements of information theory*. 2nd ed. Wiley-Interscience, 2006. ISBN: 9780471241959.

[10] Philip Dames. "Multi-robot active information gathering using random finite sets". In: (2015).

[11] Ben Grocholsky et al. "Cooperative air and ground surveillance". In: *IEEE Robotics & Automation Magazine* 13.3 (2006), pp. 16–25.

[12] Geoffrey A Hollinger and Gaurav S Sukhatme. "Sampling-based robotic information gathering algorithms". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1271–1287.

[13] Yiannis Kantaros et al. “Asymptotically Optimal Planning for Non-Myopic Multi-Robot Information Gathering.” In: *Robotics: Science and Systems*. 2019.

[14] Chris Kreucher, Keith Kastella, and Alfred O Hero. “Multitarget tracking using the joint multitarget probability density”. In: *IEEE Transactions on Aerospace and Electronic Systems* 41.4 (2005), pp. 1396–1414.

[15] Chris Kreucher, Keith Kastella, and Alfred O Hero Iii. “Sensor management using an active sensing approach”. In: *Signal Processing* 85.3 (2005), pp. 607–624.

[16] Chris Kreucher, Keith Kastella, and Alfred O. Hero Iii. “Sensor management using an active sensing approach”. In: *Signal Processing* 85.3 (2005), pp. 607–624. DOI: `10.1016/j.sigpro.2004.11.004`.

[17] Ronald Mahler. “PHD filters of higher order in target number”. In: *IEEE Transactions on Aerospace and Electronic systems* 43.4 (2007), pp. 1523–1543.

[18] Ronald PS Mahler. “Multitarget Bayes filtering via first-order multitarget moments”. In: *IEEE Transactions on Aerospace and Electronic systems* 39.4 (2003), pp. 1152–1178.

[19] G. Mathew, A. Surana, and I. Mezić. “Uniform coverage control of mobile sensor networks for dynamic target detection”. In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 7292–7299.

[20] George Mathew and Igor Mezić. “Metrics for ergodicity and design of ergodic dynamics for multi-agent systems”. In: *Physica D: Nonlinear Phenomena* 240.4-5 (2011), pp. 432–442. DOI: `10.1016/j.physd.2010.10.010`.

[21] Anastasia Mavrommati et al. “Real-time area coverage and target localization using receding-horizon ergodic exploration”. In: *IEEE Transactions on Robotics* 34.1 (2017), pp. 62–80.

[22] Lauren M Miller and Todd D Murphey. “Optimal planning for target localization and coverage using range sensing”. In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2015, pp. 501–508.

[23] Lauren M Miller and Todd D Murphey. “Trajectory optimization for continuous ergodic exploration”. In: *2013 American Control Conference*. IEEE. 2013, pp. 4196–4201.

[24] Lauren M. Miller et al. “Ergodic Exploration of Distributed Information”. In: *IEEE Transactions on Robotics* 32.1 (Feb. 2016), pp. 36–52. ISSN: 1552-3098, 1941-0468. DOI: `10.1109/TRO.2015.2500441`.

[25] Branko Ristic and Ba-Ngu Vo. “Sensor control for multi-object state-space estimation using random finite sets”. In: *Automatica* 46.11 (2010), pp. 1812–1818.

[26] Shun Taguchi and Kiyosumi Kidono. “Exclusive Association Sampling to Improve Bayesian Multi-Target Tracking”. In: *IEEE Access* 8 (2020), pp. 193116–193127.

[27] Sergio Verdú. “α-mutual information”. In: *2015 Information Theory and Applications Workshop (ITA)*. IEEE. 2015, pp. 1–6.

[28] Ba-Ngu Vo, Sumeetpal Singh, Arnaud Doucet, et al. “Sequential Monte Carlo implementation of the PHD filter for multi-target tracking”. In: *Proc. Int’l Conf. on Information Fusion*. 2003, pp. 792–799.