Autonomous Robots manuscript No.
(will be inserted by the editor)

Multi-Agent Informed Path Planning Using the Probability Hypothesis Density

Jonatan Olofsson · Gustaf Hendeby · Tom Rune Lauknes · Tor Arne Johansen

Received: date / Accepted: date

Abstract An Informed Path Planning (IPP) algorithm for multiple agents is presented. It can be used to efficiently utilize available agents when surveying large areas, when total coverage is unattainable. Internally the algorithm has a Probability Hypothesis Density (PHD) representation, inspired by modern multi-target tracking methods, to represent unseen objects. Using the PHD, the expected number of observed objects is optimized. In a sequential manner, each agent maximizes the number of observed new targets, taking into account the probability of undetected objects due to previous agents' actions and the probability of detection, which yields a scalable algorithm. Algorithm properties are evaluated in simulations, and shown to outperform a greedy base line method. The algorithm is also evaluated by applying it to a sea ice tracking problem, using two datasets collected in the Arctic, with reasonable results. An implementation is provided under an Open Source license.

Keywords path planning · target tracking · probability hypothesis density · phd · multi-agent

1 Introduction

In a multitude of applications and with growing utility, Unmanned Aerial System (UAS) agents have been successfully deployed in reconnaissance missions. Traditionally, these agents are either controlled manually (then known as Remotely Piloted Aerial Systems (RPASs)), or their search is automated based on one of several available search pat-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 642153, as well as the Research Council of Norway through the Centres of Excellence funding scheme, grant number 223254 – NTNU-AMOS. G. Hendeby has been funded by the Center for Industrial Information Technology at Linköping University (CENIIT). The TRI campaign was funded by the Fram Centre project *Ground-based radar measurements of sea-ice, icebergs, and growlers.*

J. Olofsson, T. A. Johansen
Center for Autonomous Marine Operations and Systems (NTNU-AMOS),
Department of Engineering Cybernetics,
Norwegian University of Science and Technology,
Trondheim, Norway.
jonatan.olofsson@liu.se

G. Hendeby
Department of Automatic Control,
Linköping University, Linköping, Sweden.

T. R. Lauknes
Norut, Tromsø

terns (IAMSAR 2010). While the sensor payload may vary, the arguably most common sensor type is the optical camera. With more or less effort, detections can be extracted for information value far beyond that of the raw imagery. Though sometimes adversarial, the search is often one-sided, and the detections of an agent simply depend on how the agent moves, its Field-of-View (FOV) and its detection resolution. These detections, created offline or online during flight, can be transformed to geodetic coordinates and further processed. By comparing detections with preexisting ones from the region and attempting to associate the new detections with the old, it is possible to hypothesize around further properties such as velocity and long-term trajectories, using tracking algorithms.

When planning future flights, this information can be exploited to propose flight paths which aspire to be more rewarding than the non-informed search patterns. These paths can be followed by a pilot or autonomously — although even in autonomous mode it is common for a pilot to be in the decision loop, confirming flight routes before execution. As an example from our background; given rough data from, e.g., satellite imagery, ships operating in ice-infested waters could launch UAS agents to improve the accuracy and recency of sea ice tracking around the ship. This has the potential to raise the level of safety and improve the availability of such waters for transportation routes. The map would provide a better idea of where the sea ice concentrations are higher, allowing the priority to be on those areas.

The level of reward from a given flightpath is, obviously, a question of the definition of reward. Different rewards will yield different optimal paths and some will be better suited for some missions than others. A contribution of this article is to discuss the choice of a reward function formed using a concept which is of central importance to several recent advances in target tracking.

In the past decade, target tracking and Multiple Target Tracking (MTT) has seen the exploration of the Probability Hypothesis Density (PHD). Defined as the density of the expected number of targets, we propose that the PHD is a natural candidate to form the base of a reward-function for Informed Path-Planning (IPP). Using this measure in IPP, paths can be generated to maximize the expected number of targets to be observed during the full duration of all agent's flights, i.e., the integral of the previously unobserved PHD over the observed region. Further, we derive an approximation which enables and justifies sequential computation of the paths of multiple agents, reducing the complexity of computation and implementation.

In short, the contributions of this article are:

- a novel PHD based reward function for IPP;
- a decoupled approximative reward function for the approximately static case that allows for computationally efficient multi-agent IPP;
- an algorithm description for the practical implementation of the above mentioned reward function;
- a suggestion of how to solve the optimization problem by sampling full path proposals; and
- a description of how to apply the method in an actual application, which is also used to evaluate the method.

In Sect. 3 we begin with presenting the PHD — how it is obtained and how it can be seen as a natural interface between a tracker and a path planner. In Sect. 4, we introduce how to incorporate this measure into a path planner. Implementation considerations are presented in Sect. 5. In Sect. 6, we present two real-world scenarios where sea ice objects are tracked off the coast of Svalbard, and flight plans are created for a number of reconnoitering missions with multiple UASs. Finally, discussion and concluding remarks are presented in Sect. 7 and Sect. 8 respectively.

2 Background

Here we provide a background to the article, including some of the terminology used in the article and a detailed formulation of the problem addressed.

2.1 Terminology and Notation

In this article, a set of agents $\mathcal{A} \ni a$ is considered, where the α'th agent in the ordered set is referred to as a_α. Variables associated with specific agents are subscripted with a or a_α, although this subscript may be dropped when speaking generally.

The agents observe and report observations from a region around their current position — their FOV. A target, located at position $\boldsymbol{x}$, is observed with probability $\bar{p}_{d,a}(\boldsymbol{x})$ by agent a, if and only if $\boldsymbol{x}$ is within the FOV $S_a(\boldsymbol{x}_a)$ of agent a at position $\boldsymbol{x}_a$. This can be expressed with a varying probability of detection,

$$p_{d,a}(\boldsymbol{x}) = \begin{cases} \bar{p}_{d,a}(\boldsymbol{x}), & \boldsymbol{x} \in S_a(\boldsymbol{x}_a) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Trajectories are represented as discrete series of waypoint coordinates $\mathcal{P}_{k,0:K,a}$. These paths, for all agents, are included in the set $\mathcal{P}_{k,0:K}$. Their respective subscripts indicate the paths start at time k, and contains positions for each discretized timestep $k+\kappa$, $\forall \kappa \in [0,\ldots,K]$ where K is the planning horizon. Unless needed, the leading k index may be dropped and assumed to be the start of the plan (i.e. "now"). For distinct times κ, the set of coordinates of all agents is denoted $\mathcal{P}_{k,\kappa}$.

2.2 Problem Formulation

In IPP, the main objective for path planning is to maximize a reward function under certain constraints — mathematically formulated as the following high-level equation:

$$\mathcal{P}^* = \operatorname*{arg\,max}_{\mathcal{P} \in \Psi} I(\mathcal{P}) \quad \text{s.t.} \quad c(\mathcal{P}) \leq \beta \tag{2}$$

where Ψ is the set of possible and permissible trajectories for one or more agents; $I(\mathcal{P})$ is the information quality measure along the trajectory $\mathcal{P}$; $c(\mathcal{P})$ is the cost associated with the path, and β is the budget constraint. The limitations of Ψ can also include, e.g., geographical constraints (geofencing), velocity limits (maximum/minimum distance between consecutive points) as well as other constraints.

Seeking to maximize the expected value of a probabilistic reward function R we let, for a set of discretized trajectories $\mathcal{P}_{0:K}$,

$$I(\mathcal{P}_{0:K}) = \mathbb{E}[R(\mathcal{P}_{0:K})]. \tag{3}$$

In our application, $\mathcal{P}_0$ — the starting points of all agents — is predetermined although this is not a necessary requirement.

In particular, this objective function is tailored for a scenario in which we have a finite amount of flight time for one or more agents, and we wish to propose a flight plan based on the available tracking information. Using the PHD as the basis for the reward function we seek to maximize the *expected number of targets detected by the agents* throughout a fixed flight-time. Relevantly, this measure is a central part of recent advances in MTT, as described in Sect. 3. While there are numerous performance metrics available for path planning, we argue that its availability along with its property of maximizing the number of detected targets — or equivalently, minimizing the number of undetected targets — makes it a compelling choice in many scenarios.

A visual illustration of the effect on the PHD of unobserved targets from observing a part of a region is given in Fig. 1. The residual PHD describes a lower expected number of undetected targets than the surrounding region, and is thus seen as a seemingly cut-out region. The optimal set of paths that minimize the unobserved PHD will also maximize the cut-out volume. Another, more formal, illustration of the effect of an observation is given in Sect. 4.1 and Fig. 2.

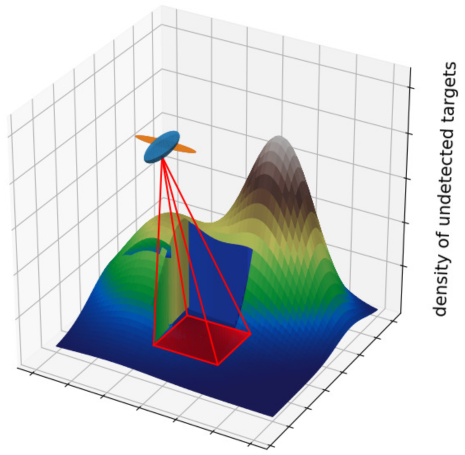

Fig. 1 PHD of expected number of undetected targets as affected by a single observation

2.3 Related work

This article combines ideas from two major fields: Multiple Target Tracking and Informed Path-Planning, both with major bodies of available literature.

2.3.1 The Probability Hypothesis Density

The use of the PHD (Winter and Stein 1993) for MTT was popularized in MTT in the early 00's by Mahler (Mahler and Zajic 2001; Mahler 2003), who made it central in his derivation of the PHD filter. This filter has since been thoroughly expanded into several implementations (Sidenbladh 2003; Vo and Ma 2006; Pasha et al. 2006) and variants for e.g. tracking of extended targets (Granström and Lundquist 2012). Mahler (2007b) provides the foundations for notable related algorithms of MTT which — at least originally (Streit 2017) — were derived using Finite Set Statistics (FISST). These include, in order of appearance, the PHD filter (Mahler and Zajic 2001), the Cardinalized Probability Hypothesis Density (CPHD) filter (Mahler 2006, 2007a; Vo et al. 2006), the Multiple-Target Multi-Bernoulli (MeMBer) (Mahler 2007b), and Cardinality-Balanced Multiple-Target Multi-Bernoulli (CBMeMBer) (Vo et al. 2009, 2007). Later, the δ-Generalized Labeled Multi-Bernoulli (δ-GLMB) (Vo and Vo 2013; Vo et al. 2014) was introduced as a FISST variant of the Multiple Hypothesis Tracker (MHT) filter (Brekke and Chitre 2018; Williams 2015; Olofsson et al. 2017a; Reid 1979), eventually resulting in its simplified relative, the Labeled Multi-Bernoulli (LMB)-filter (Reuter et al. 2014). While none of the latter filters directly manipulate the PHD as the PHD filter, it remains easily extractable as all the relevant statistics are tracked. With sufficient assumptions, in particular regarding existence probabilities, the PHD could in theory be extracted from any MTT algorithm.

2.3.2 Informed Path Planning

IPP and the related field of adaptive sampling (Fiorelli et al. 2006) has been studied using a multitude of approaches, in varying level of detail of constraints and different measures of path quality. The goal is to generate trajectories for movable sensing agents, to maximize the value of the sampled data. Unlike Coverage Path-Planning (CPP) (Galceran and Carreras 2013), paths from IPP do not aim to cover the entire region, but is limited to maximizing the gain given a restricted budget.

Broadly, the problem can be split into the studies of *i*) the optimization algorithm yielding the actual paths; and *ii*) the reward function against which the paths are scored to determine optimality.

In (Hollinger and Sukhatme 2014), a survey is presented as well as three proposed sampling-based algorithms for optimizing a general reward function. The approach is inspired by point-to-point planning algorithms (where starting points, endpoints and obstacles are known), such as the asymptotically optimal Rapidly-exploring Random Tree* (RRT*), but expands to more generic budgetary constraints such as time, fuel, or energy.

In Skoglar (2012), methods are explored to maximize the information gain (i.e. minimizing the covariance) in a gridded map of information filters. While the optimization routine employed limits the planning horizon, the measure itself could potentially be used similarly to the PHD measure proposed in this article.

In (Singh et al. 2009), the *recursive-greedy* path planning of (Chekuri and Pál 2005) is extended to multiple agents through sequentialization.

Using the PHD for path planning is not an entirely novel idea in itself — Dames (2017) introduced a distributed PHD-based planning algorithm in which each agent locally moves towards the PHD maximum within its own Voronoi-cell. The article is expanded with a 3D example in (Dames 2019). Instead of applying the PHD filter, our approach allows for a more flexible choice of tracking algorithm — we use the LMB filter, but the planning need only that the PHD can be extracted from the tracker.

Further, unlike (Dames 2017, 2019), this article takes a centralized planning approach which finds not only destination points but a list of waypoints for the entire flight path of each agent. This also includes taking into account the effects the other agents will have on a joint reward function throughout their respective flight path.

3 Target Tracking and the Probability Hypothesis Density

The modern field of target tracking has evolved from signal filtering theory pioneered during World War II (Wiener 1965). Through a series of evolutionary steps it has seen the development of the indisputably significant Kalman filter (Kalman 1960) and later the Particle Filter (Gordon et al. 1993). Further, it has extended into the exploration of scenarios with sensors that can detect multiple sources yet not distinguish their identities, giving rise to the subfield of MTT.

3.1 Bayesian Multi-Target Tracking and the Probability Hypothesis Density

In Bayesian tracking, efforts are focused on finding a tractable solution to the general Bayesian recursion (Mahler and Zajic 2001):

$$\pi_{t|t-1}\left(x_t|z_{1:t-1}\right) = \int f_{t|t-1}\left(x_t|x_{t-1}\right)\pi_{t-1}\left(x_{t-1}|z_{1:t-1}\right)dx_{t-1} \tag{4a}$$

$$\pi_{t|t}\left(x_t|z_{1:t}\right) = \frac{g_t\left(z_t|x_t\right)\pi_{t|t-1}\left(x_t|z_{1:t-1}\right)}{\int g_t\left(z_t|x_t\right)\pi_{t|t-1}\left(x_t|z_{1:t-1}\right)dx} \tag{4b}$$

with f and g being the predictive distribution and measurement likelihood, respectively, and $\pi_{t|t-1}$ ($\pi_{t|t}$) is the target state Probability Density Function (PDF) for time t given data up until time $t-1$ (t). z_t is the set of measurements at time t, and $z_{1:t}$ that of all those registered up until time t. For practical applications, the generalized formulation of (4) is altered according to simplifying assumptions to form the various tracking algorithms, such as the prevalent Kalman filter.

The same formulation can be even further generalized to the MTT case by reconceptualizing the meaning of the involved terms (Mahler and Zajic 2001). Mahler's FISST concept of a multi-target PDF means that the involved variables come to represent distributions of sets of targets and reports — Random Finite Sets (RFSs). These probabilistic sets can not only express the uncertainty in the tracking of the individual targets, but also the uncertainty of the existence of each target.

This extension of PDFs raises the question of the interpretation of its moments. The definition of the first moment employed in FISST is known as the PHD (Winter and Stein 1993) and corresponds in each point to the density of expected set objects at that point. This density, $v(x)$, is normally integrated over a region S, giving the expected number of objects within that region:

$$V(S) = \mathbb{E}[|\mathcal{X} \cap S|] = \int_S v(x)dx \tag{5}$$

where $\mathcal{X}$ is the set of all targets, stochastic in content and cardinality. This set may or may not be explicitly known, as the measure here is the set's expected cardinality. The intersection operator indicates the windowing of each target's PHD with the region of interest.

Among the MTT algorithms, the PHD filter (Mahler and Zajic 2001; Vo and Ma 2006) and its subsequent extensions in particular incorporate this measure directly into its core. Other algorithms, such as the δ-GLMB (Vo et al. 2014), MHT (Reid 1979) and the LMB (Reuter et al. 2014) filters, stores its state as a more classical target-list. Contrary to the PHD filter, those filters do not generally implicitly consider non-detected targets, so this may have to be modeled and added as per

$$v_t = v_t^\dagger + v_t^n \tag{6}$$

for the tracker-generated PHD $v_t^\dagger$ and the model for non-detected target, v_t^n. The comparative values of the tracked and modeled PHD expressed in (6) can be used to balance between *exploration* and *exploitation* in the region of interest.

When observing a region S, for which a PHD has been constructed, the expected number of observed targets is

$$\mathbb{E}[|\mathcal{X}_{obs}|] = \int v(x) \cdot p_d(x)\, dx \tag{7}$$

for a sensor with probability of detection $p_d(x)$. Conversely, the expected number of undetected targets after a single observation is given by

$$\mathbb{E}[|\mathcal{X} \setminus \mathcal{X}_{obs}|] = \int v(x)(1 - p_d(x))\, dx. \tag{8}$$

By extension, the expected number of targets to be missed despite multiple observations $g \in G$ (and respective probability of detection $p_{d,g}(x)$) is

$$\mathbb{E}[|\mathcal{X} \setminus \mathcal{X}_{obs}|] = \int v(x) \cdot \left[\prod_g (1 - p_{d,g}(x))\right] dx. \tag{9}$$

Observations g may or may not be from separate agents.

4 Informed Path Planning

Given limited resources, we seek to maximize the *expected number of targets detected by the agents*, using the PHD as foundation.

When studying this measure more closely, and consider that multiple agents observe the region on potentially multiple occasions, the question arises which agent or observation contributes with what. Given the optimization objective, this question has an intrinsic dependency on the order of the observations as, given equal probability of detection, most targets are likely to be first detected on the first viewing. Plans are hypothetical however, and if the first observation is skipped due to change of plans, the seminal higher score would instead be attributed to another agent first to observe the region. In this section, we seek to formulate and approximately decouple this dependency in the agents' path plannings and formulate sequential reward functions where agents, in ordered turn, commit to paths that optimize the remaining reward. This allows us to formulate the reward as a sum of the contributions of each agent:

$$R(\mathcal{P}_{0:K}) = \sum_{a \in \mathcal{A}} R_a(\mathcal{P}_{0:K,a}) \tag{10}$$

where R_a is the reward obtained by agent a over its path $\mathcal{P}_{0:K,a}$.

4.1 Complete reward function

Importantly, (10) is a submodular function (Krause and Guestrin 2011), meaning that for two example paths q and w, $I(\mathcal{P}_{q+w}) \leq I(\mathcal{P}_q) + I(\mathcal{P}_w)$, i.e.: The reward for executing both paths will be less than or equal to that of their individual execution, as the reward gain is less if any of the targets detected in an observation has been previously observed by another agent.

To begin formulating our proposed reward, we introduce $\rho(\boldsymbol{x})$, which represents the reward density at the point $\boldsymbol{x}$ given the set of all observations G. Each observation has a detection probability $p_{d,g}(\boldsymbol{x})$. As we are only interested in disjoint detections, we note that the number of individual detected targets is governed by the inclusion-exclusion principle:

$$\rho(\boldsymbol{x}) = v(\boldsymbol{x}) \sum_{i=1}^{|G|} \left((-1)^{i-1} \sum_{\Gamma \subset G, |\Gamma| = i} \prod_{g \in \Gamma} p_{d,g} \right) \tag{11}$$

where the alternating negative sign ensures targets are only counted once (compare: $|\mathcal{A}| \cup |B| = |\mathcal{A}| + |B| - |\mathcal{A}| \cap |B|$).

If observations are assumed to be sequential, the value added from each observation can be seen by rearrangement of (11) to recursive form (Chen 2014):

$$\varpi_0 = 0 \tag{12a}$$

$$\varpi_i(\boldsymbol{x}) = p_{d,a_i}(\boldsymbol{x}) \left(1 - \sum_{\kappa=1}^{i-1} \varpi_\kappa(\boldsymbol{x}) \right) \tag{12b}$$

$$\rho(\boldsymbol{x}) = v(\boldsymbol{x}) \sum_{i=1}^{|G|} \varpi_i(\boldsymbol{x}), \tag{12c}$$

where ϖ_i is the probability that a target is first observed by observation i.

The interpretation of (12) is that only reports from previously undetected targets will add to the reward function, i.e. the reward of an agent's path is only dependent on the yet undetected targets. Note however that there is an inherent dependency introduced by the ordering of G, not only in time but also against simultaneous observations by other agents.

In the time dimension, running all agents in parallel, we seek to expand R into separate parts for each timestep:

$$R(\mathcal{P}_{k,0:K}) = \sum_{\kappa=0}^{K} R(\mathcal{P}_{k,\kappa} | \bar{v}_{k,\kappa}), \tag{13}$$

where $\bar{v}_{k,\kappa}$ is the PHD of yet unobserved targets, extracted at time k but affected by all agents following their planned paths up until planning time $k+\kappa$, $\mathcal{P}_{k,0:\kappa}$. Notably, this formulation retains the (causal) submodularity property through the conditioning of R on $\bar{v}_{k,\kappa}$.

Here we limit the discussion to $v_{k+\kappa} = v_k \quad \forall \kappa \in [0, \ldots, K]$, i.e. the environment is assumed static for the duration of the plan allowing us to drop the k index unless needed. Treating all agents jointly, the unobserved PHD can then be recursively defined as

$$\bar{v}_0 = v_k \tag{14a}$$

$$\begin{aligned} \bar{v}_{\kappa+1} &= \bar{v}_\kappa \cdot \prod_{a \in \mathcal{A}} (1 - p_{d,a}(\boldsymbol{x})) \\ &= \bar{v}_\kappa \cdot (1 - p_d(\boldsymbol{x}))^{\mathcal{A}}. \end{aligned} \tag{14b}$$

With this definition, using the multi-object exponential notation as above, we define

$$R(\mathcal{P}_{k,\kappa} | \bar{v}_{k,\kappa}) = \int \left[\left(1 - (1 - p_d(\boldsymbol{x}))^{\mathcal{A}} \right) \cdot \bar{v}_\kappa \right] dx \tag{14c}$$

to obtain the desired reward function. The parts of the original PHD v that these quantities represent are visualized for a single observation in Fig. 2. In the example underlying Fig. 2, the observed region has a uniform probability of detection. This yields a PHD of unobserved targets that is, within the FOV, simply a downscaled version of the full PHD. Non-uniform probabilities of detection will yield a more complex shape.

4.2 Agent Decoupled Reward Function

The recursion in (13) and (14) notably depends on the full solution $\mathcal{P}_{k,0:K}$ of paths for all agents. For an optimization solver, this translates to a significant increase of the search space for each additional agent, which in many cases will make the problem tractable for only a small number of agents. To reduce the dimensionality of the optimization problem and simplify the implementation, we seek to decouple the planning of each agent's path $\mathcal{P}_{0:K,a}$ from the others'. For this, we return to the ordering of the observations implied in (12) and posit that each agent's full path is rewarded in consecutive sequential order. Some of the implications of this have already been noted, such that the reward becomes less for subsequent observations, making the reward conditioned on the order that the paths are planned. The ordering entails that the path planning for the set of agents can be made sequentially while maintaining a recursive description of the unobserved PHD:

$$\bar{v}^{1}_{(-1)}\left(\boldsymbol{x}\right) = v_k\left(\boldsymbol{x}\right) \tag{15a}$$

$$\bar{v}^{\alpha}_{(-1)}\left(\boldsymbol{x}\right) = \bar{v}^{\alpha-1}_{K}\left(\boldsymbol{x}\right) \tag{15b}$$

$$\bar{v}^{\alpha}_{\kappa}\left(\boldsymbol{x}\right) = \bar{v}^{\alpha}_{\kappa-1}\left(\boldsymbol{x}\right)\cdot\left(1-p_{d,a_\alpha,\kappa}\left(\boldsymbol{x}\right)\right) \tag{15c}$$

$$\bar{V}^{\alpha}_{\kappa}\left(S\right) = \int_S \bar{v}^{\alpha}_{\kappa}\left(w\right)dw \tag{15d}$$

$$R_\alpha\left(\mathcal{P}_{0:K,a}|\bar{v}^{\alpha-1}_{K}\right) = \sum_{\kappa=0}^{K}\bar{V}^{\alpha}_{\kappa}\left(S_{a_\alpha}\left(\mathcal{P}_{\kappa,a_\alpha}\right)\right) \tag{15e}$$

$$R\left(\mathcal{P}_{0:K}\right) = \sum_{\alpha=1}^{|\mathcal{A}|} R_\alpha\left(\mathcal{P}_{0:K,a_\alpha}|\bar{v}^{\alpha-1}_{K}\right). \tag{15f}$$

where $\bar{v}^{\alpha}_{\kappa}$ is the unobserved PHD prior to observations by agent α at time κ.

The equations can be described as follows:

- (15a) — initial PHD;
- (15b) — next agent's initial PHD;
- (15c) — next unobserved PHD (next timestep);
- (15d) — expected number of targets within FOV;
- (15e) — agent's total reward; and
- (15f) — total reward.

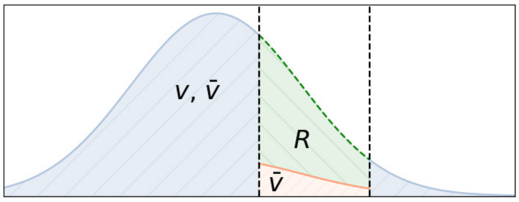

Fig. 2 Parts of the PHD corresponding to used symbols

As indicated by the conditioning in (15f), the full path is now optimized based on the PHD of targets unobserved after the previous agent has committed to its path. This allows the optimization routine to parallelize by generating independent samples of the agent's full path which can be scored and ordered according to the reward function. A pseudocode implementation of (15) is exemplified in Alg. 1. The optimization step in the algorithm is further discussed in Sect. 5.2.

Algorithm 1 Sequential PHD Path Sampling

```
$\bar{v}\left(\boldsymbol{x}\right) \leftarrow v\left(\boldsymbol{x}\right)$
for $a \in \mathcal{A}$ do
    $\mathcal{P}^{a}_{0:K} \leftarrow \arg\max_{\mathcal{P}\in\Psi}\mathbb{E}\left[R\left(\mathcal{P}|\bar{v}\right)\right]\text{ s.t. }c\left(\mathcal{P}\right)\leq\beta$
    $\bar{v}\left(\boldsymbol{x}\right) \leftarrow \bar{v}\left(\boldsymbol{x}\right)\prod_{\kappa=0}^{K}\left(1-p_{d,a,\mathcal{P}_\kappa}\left(\boldsymbol{x}\right)\right)$
end for
```

5 Implementation

To demonstrate the practical application of the planner reward function proposed in Sect. 4, a tracker and a proof of concept planner were implemented[1]. The planner is a genetic algorithm with Monte Carlo-generated

[1] The LMB tracker used is available as Free and Open Source Software (FOSS) at `https://github.com/jonatanolofsson/clmb`. The planner described in this chapter is available at `https://github.com/jonatanolofsson/phdplanner`.

proposal paths, but more elaborate algorithms can be found in, e.g., (Hollinger and Sukhatme 2014). This section details practical considerations of the implementation.

The tracker and planning algorithms are implemented in C++, with a Python interface using the pybind11 library. Parallelization of the C++ algorithms is implemented using the OpenMP API.

5.1 Coordinate Systems and PHD Sampling

In the implementation of the path planning, we assume the PHD to be sampled in an axis-aligned Cartesian region. In our examples in Sect. 6, this translates to a latitude-longitude aligned gridded linearization of the observed area. After the latitude-longitude area has been selected, the tracked targets in that region are selected and transformed, along with the region and all the reports, into a Cartesian North-East (NE) system with origin in the region centerpoint. Monte Carlo-integration is then performed by sampling N_s points from each targets' positional distribution. The points from each target $\ell \in \mathcal{L}$ is then binned into a square grid, where r_ℓ / N_s is added to the value of each cell for each point that falls within that bin, making the total added value by all points $\sum_{\ell \in \mathcal{L}} r_\ell$.

The grid cells in the sample implementation represent an area which approximates a single observation scan, such that the step-size for each planning step is one cell in either direction. A finer grid, paired with more elaborate FOV calculations and step generation would provide a better estimate of the final path reward, but would also extend the complexity and memory requirements of the algorithm significantly which, in the intended application, is undesirable.

5.2 Sampling Paths

In a Monte Carlo-based path planning, a central piece is the generation of samples from the set of valid paths. This set should be tailored to the application to take, e.g., turn-rate restrictions into account. In this specific application, this set is that of all paths of exactly K steps to neighboring cells.

To generate random samples from this set, we sample informative points $\overrightarrow{\boldsymbol{x}_S}$ from the remaining unobserved PHD, and fly straight towards that point until the point is reached, or the step-limit is reached. If there are steps remaining, a new point is sampled and the process repeated.

The normalized unobserved PHD is chosen as the point proposal distribution, gridding S into cells S_i:

$$\overrightarrow{\boldsymbol{x}_S} \sim \frac{v_\kappa^\alpha(\boldsymbol{x})}{\int_S v_\kappa^\alpha(\boldsymbol{x})\, d\boldsymbol{x}} \cong \frac{V_\kappa^\alpha(S_{\boldsymbol{x}})\,\delta_c(\boldsymbol{x})}{\sum_i V_\kappa^\alpha(S_i)}. \tag{16}$$

In the approximation expressed in (16), $S_{\boldsymbol{x}}$ is the cell in which the $\boldsymbol{x}$ is located and $\delta_c(\boldsymbol{x}) = \sum_i \delta_{c_i}(\boldsymbol{x})$ is the sum of Dirac delta functions centered around the cell centerpoints. The Dirac delta sum makes the probability distribution of $\overrightarrow{\boldsymbol{x}_S}$ nonzero at the cell centerpoints only. The values at those centerpoints correspond to the PHD integrated over the area of that cell. The values of this discrete distribution can be stored as an array, making its Cumulative Distribution Function (CDF) relevantly easily invertible for sampling.

Each sampled point adds to the generated path by, for each time step, moving the agent to an adjacent valid cell's centerpoint towards the sampled point. The resulting proposal path can be scored according to (15d).

After generating $N_{\mathcal{P}}$ proposal paths, the highest scoring path is selected as the proposed flight path for that agent, and the resulting remaining unobserved PHD is used for the planning of the next agent's path.

This Monte Carlo algorithm can be utilized as the first step of a genetic algorithm (Mitchell 1996). This is an easy-to-implement optimization routine which combines existing "generations" of a population of solutions — favoring high scoring solutions — to create new, potentially better, solutions.

Experiments, however, showed that generating a larger initial population far outperformed — performance-wise and reward-wise — any benefits of breeding subsequent generations, though this may be subject to future improvements.

In Sect. 6 we also, in addition to the Monte Carlo-algorithm, compare results with a "greedy" agent, which instead of randomly sampling informative points always heads for the maximum point in the unobserved PHD.

5.3 Simulated Tests

To quantify the benefit of the proposed optimization measure in a controlled environment, a simulation was set up in which 40 reports were uniformly randomly distributed in a 200×200 m^2 area. A covariance of $30^2 \cdot \boldsymbol{I}_2$ m^2 was associated with each of the reports, which were then fed to the LMB tracker. The PHD from the tracker, static in the simulation, forms the background of Fig. 3, where also the resulting paths are overlaid. The figure shows the randomly distributed set of tracked targets, and the paths generated by the Monte Carlo (blue) and greedy (orange) path-generation respectively. The resulting rewards were also compared to that of a lawnmower pattern search. For a fair comparison, all agents were set to start from the lower left corner. The lawnmower agent was moving bottom-to-top, left-to-right. Note the behaviour that with no modeling of undetected targets, the planning focuses solely on reconfirming the existing targets.

In Fig. 4, the expected number of detections for the generated paths were compared. In this scenario, each agent operated independently with individual accumulation of expected target detections. The results show that even with simple optimization routines, such as described in this section, the proposed measure quickly outperforms the baseline lawnmower pattern in terms of the expected number of detected targets. In this simulated environment, the greedy path-generation generally performs similarly to the Monte Carlo one. On a 2015 laptop, 200,000 Monte Carlo paths of 1000 steps were constructed and evaluated in approximately 13 s (approx. 64 µs per path), whereas the greedy method — generating only a single path — required 0.5 ms.

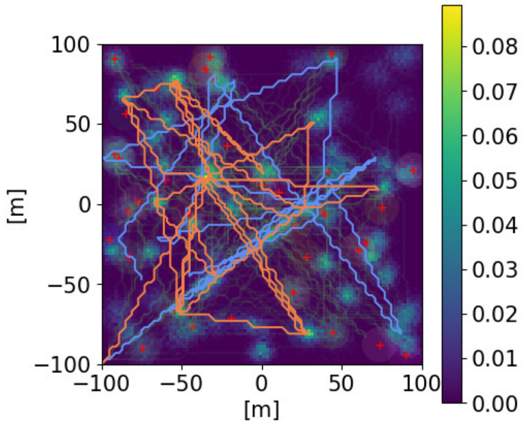

Fig. 3 Paths generated by Monte Carlo (blue) and greedy (orange) algorithms. The value of the PHD is given as background, and measurement centerpoints as red marks

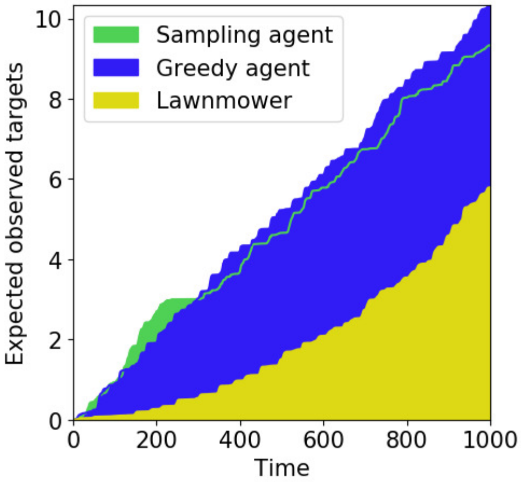

Fig. 4 Individual accumulation of expected target detections

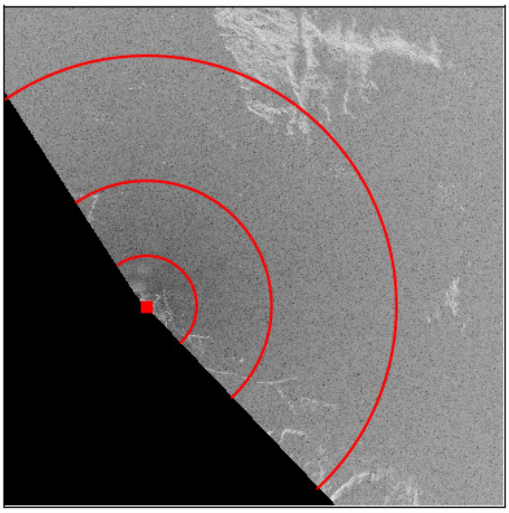

Fig. 5 Example of a radar scan from the TRI sensor, with brighter areas of sea ice and an overlaid illustration of the radial nature of its raw measurements

6 Applications

In this section we describe how the proposed methods were applied to maximize the number of future sea ice detections based on two separate real-world datasets, and the considerations that come with different types of data. Both datasets were captured by the fjord Kongsfjorden, Svalbard, where ice breaks off the nearby glaciers and flows towards the ocean. The data was collected in an ongoing effort to study and predict sea ice movements in order to, e.g., protect ships and permanent installments in the Arctic. The Terrestrial Radar Interferometer (TRI) image in Fig. 5 gives a general overview of the area, as observed by a TRI sensor placed in Kongsfjorden during a 2018 field campaign. The dataset recorded by this sensor is used in the first application, but the area also roughly corresponds to the area surveyed in the 2017 UAS flights used in the second application. In each scenario, we wish to send out two additional sensors carried by UAS agents to provide additional data to be registered in the map.

In both applications, the detections were tracked in an LMB tracker as detailed in (Olofsson et al. 2017b). The PHD was extracted from the tracker at the latest timestep and used for the generation of path proposals, as illustrated in Fig. 6. The model for the density of unobserved targets, v^n, is added to the PHD of the tracker and forwarded to the IPP module. The result of the planning can then be given either directly to the agents (UAS agents in this example) or to pilots for confirmation and execution. The observations performed by the agents by following the proposed path is fed back to the tracker, thereby closing the loop and using the data to update or propose later paths.

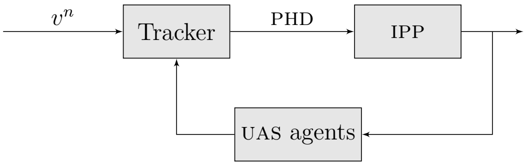

Fig. 6 Path planning system overview

In both applications, the PHD is considered static with respect to the planning timeline, and the static equations from Sect. 4 were applied. The Monte Carlo path-generation are in both cases compared to the greedy search explained in Sect. 5.2, as an additional example of how the measure can be employed.

6.1 Terrestrial Radar Interferometer Dataset

The first dataset is part of a long-term monitoring effort with a TRI monitoring primarily sea ice drift in the fjord. The dataset was gathered in spring 2018, with images taken in 15 minute intervals. The data from the radar scans is used to form a tracking map of the region. The radar is located at the red dot in Fig. 5, and images are detected in a range-angle coordinate system.

The raw data from the TRI was transformed to geocoded Cartesian images with 5m × 5m resolution. The series of images were then processed using the OpenCV MOG2 background subtractor (Zivkovic 2004) and extracted from regions with a threshold of a minimum of 10 connected pixels. Uniformly across the scanned region, each detection was assigned a covariance of $30^2\,\mathrm{m}^2 \cdot \boldsymbol{I}_2$ to form Gaussianly distributed reports. The PHD was sampled with a gridsize of $200\,\mathrm{m} \times 200\,\mathrm{m}$, giving a 100×100 grid.

From repeated observation of the whole region, tracks were formed with an LMB tracker, with tracking data shown in Fig. 7. In Fig. 8, plans are shown for two collaborative agents. Agent 0 (green) will start its observations at coordinates $(4000, 4000)$ whereas agent 1 (blue) will begin at $(5000, 30)$. Both agents will make observations for 600 timesteps. The same scenario was also run using greedy path-generation, displayed in Fig. 9. Notably, using the greedy path-generation, the storage order and max-search algorithm is visible in planned paths, as it yields straight vertical paths with a tendency to ignore gains from visiting nearby neighbors. This is because equivalently valued maxima are selected in order bottom-to-top, left-to-right, mirroring the order the elements are stored in memory.

For the two sets of collaborating agents respectively — the Monte Carlo agents and the greedy agents — cumulative rewards are compared in Fig. 10. The cumulative rewards show that the majority of the observational value is attributed to the first agent, leaving less to the second agent. In this scenario, the agents cooperate, adding to a common expected number of observed targets. The Monte Carlo optimization was run with $1,000,000$ path proposals per agent at a cost of approximately 40 µs per path (giving a total of approximately 85 s in total), whereas the greedy path was generated in approximately 0.1 s per path (0.2 s in total).

6.2 Fixed-wing UAS dataset

The second dataset was recorded during a campaign in 2017, using the Cryowing Scout fixed wing UAS developed by Norut. In the first run — plotted in Fig. 11 — one agent flew a fixed pattern predefined by the pilot using manual waypoints. The observations made during this flights forms one of the bases for the total PHD of the region. To emphasize the modeling possibilities available in the proposed measure through v^n, we also add a simulated flow of sea ice, running from the north-east corner of the region to the west. In Fig. 11, this is visible as a higher density of targets expected in the shown area.

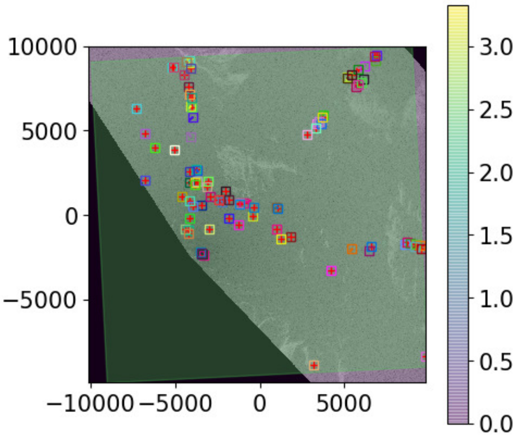

Fig. 7 Sea ice objects, represented by uniquely colored squres, were tracked using an LMB tracker

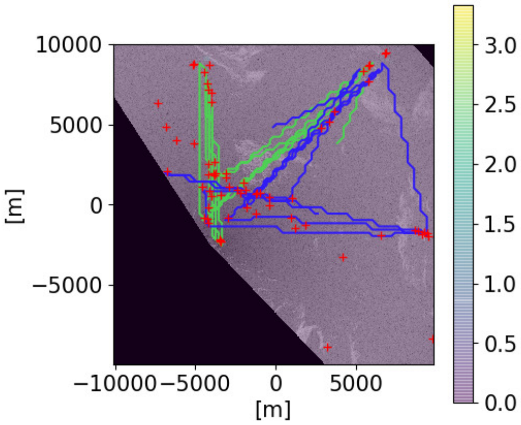

Fig. 8 Monte Carlo planning for two agents the TRI dataset

Apart from optical imagery, the flight data includes position data, as well as FOV. The data from which the tracking estimate is constructed is recorded from a downwards facing camera, with images taken every third second of the flight. Together with speed of flight, FOV

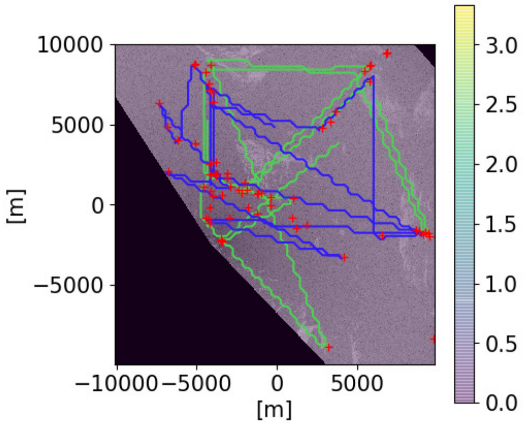

Fig. 9 Greedy path planning for two agents the TRI dataset

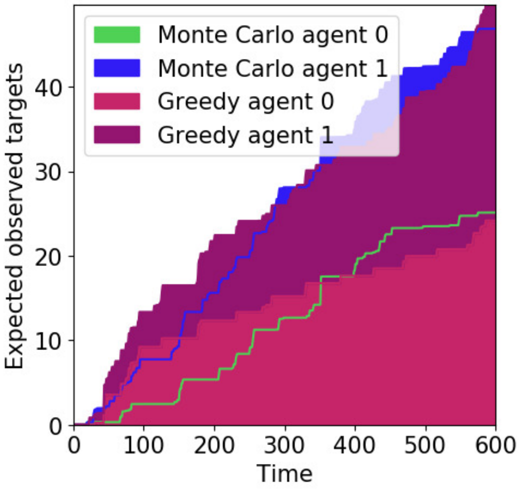

Fig. 10 Cumulative rewards for the two sets of collaborative agents in the TRI dataset

and mission altitude, this entails that a detected object is within FOV for an approximate maximum of three frames. The object detection algorithm is performed online and the results stored in a SQL database from which the flight can be replayed.

The tracking from this dataset is severely hampered by the limited time an object is within view and the lack of revisits. This makes velocity estimates of the sea ice objects very uncertain. This limits the temporal persistence of the value of the observations, prompting

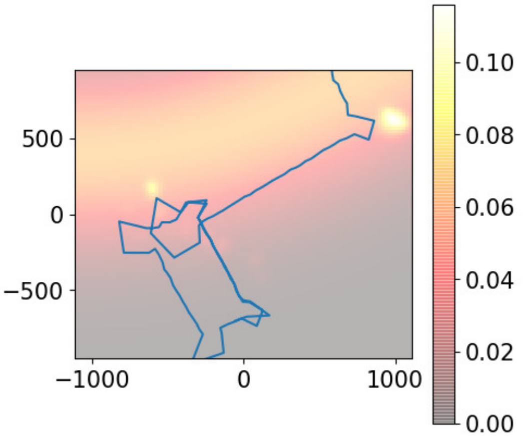

Fig. 11 Data collection flight pass with resulting PHD and model for the PHD of undetected targets

for faster use of the data. Further, the limited FOV significantly increases the dependency on the model for undetected targets, as most of the region is out of view most of the time, and was only partially observed in the initial flight. Fig. 11 shows the flight-path for the initial data-collection together with the modeled stream. The planned paths using Monte Carlo and greedy path-generation is displayed in Fig. 12 and Fig. 13 respectively — each for two collaborative agents. The path planned for the two agents are optimized sequentially. Similarly to the TRI example, the greedy equivalent of the scenario in Fig. 12 shows a less random behaviour, albeit arguably with a lesser diversification to compensate for uncertainties in the model. Just as in the TRI application, the Monte Carlo optimization was run with $1,000,000$ path proposals per agent at a cost of approximately 40 µs per path (in this scenario, 85 s in total), whereas the greedy path was generated in approximately 0.1 s per path (0.2 s in total).

The cumulative rewards for both path-generation approaches are compared in Fig. 14. The gain in cumulative reward in a collaborative effort of multiple agents is, compared to the first application in Fig. 10, more evident in this kind of map, where the PHD is more spread out.

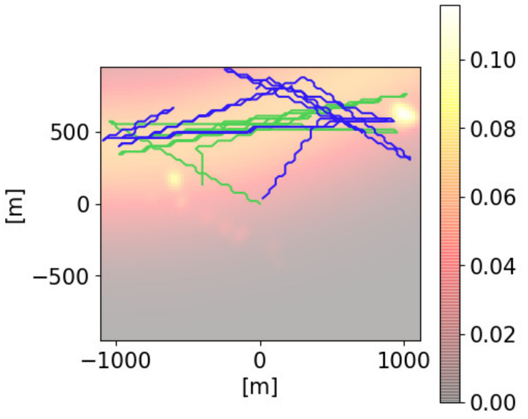

Fig. 12 Monte Carlo path-generation for two agents in the UAS dataset

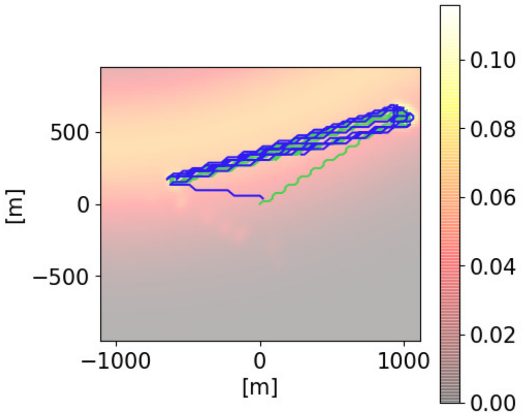

Fig. 13 Greedy path-generation for two agents in the UAS dataset

7 Discussion

Interesting insights can often be found at the intersection between research fields. While the fields of path planning and target tracking are closely related and naturally combinable, new findings in one can be continuously synergized with that of the other. The PHD can provide a new common language between the fields and, when augmented with a model for undetected targets, a natural way to express a map of the trade-off between exploration and exploitation.

The PHD as a measure is further of particular interest, as it is an integral part of the emerging family of RFS tracking algorithms. The two applications demonstrate the possibilities of the measure for practical purposes, and the implementation shows that even a simple planner can outperform the uninformed case when given a limited time. For many applications, a simple greedy planner may give sufficient behaviour, especially when speed is an important factor. It will, however, likely fall out of grace as more constraints are added to the problem, as it lacks the tools to accommodate more than the simplest of constraints. While the Monte Carlo-solver performs significantly slower in the examples, its speed is tunable — at the cost of path quality — and can even be gracefully terminated after a given per-agent time-limit.

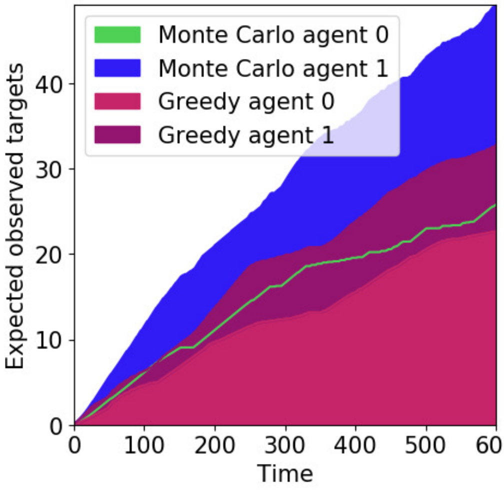

Fig. 14 Cumulative reward for two sets of collaborating agents in the UAS dataset, using Monte Carlo and greedy path-generation respectively

The use of a PHD measure proposed in this article was presented for the static cases of IPP. The decoupling presented provides a way to efficiently scale the planning to multiple agents, but also has an additive property in that should — say — only the three best out of four plans be carried out, the measure and the optimized paths will still be valid for those three agents.

It is worthy of note that the measure solely optimizes the expected number of detected targets. Other measures may, e.g., better take into consideration the fact that additional information, such as velocity and other not directly observable states, will require revisits and additional measurements over time, something that is partially discouraged in order to maximize the number of detections. However, unlike e.g. Dames (2019), the exploration is exhaustive given enough flight time as the flight plan does not merely stop at the maximum but continues to search less weighted areas. This can be summarized as a difference between short-term planning (e.g. Dames (2019)) and long-term planning (as used in this article). Just like how, e.g., Model Predictive Control (MPC) is applied, the long-term plans can always be converted to short-term plans by truncation whereas the opposite conversion of course is impossible. Short-term solutions thus trades any claim on global optimality against computational benefits.

Weighted combinations of different measures can provide a calibratable trade-off to suit applications. The combination of other measures and the application of other optimization routines is a natural extension of this work in future research. So too is the application of the algorithm in more dynamic environments and simulations, and the more general question of applying PHD-based planning in a dynamic environment. This would, e.g., include research on how often plans would need to be updated with respect to different target dynamics and agility.

8 Conclusion

A method for multi-agent Informed Path-Planning (IPP) has been presented. It can be used to better utilize available resources in surveillance situations, in particular situations where exhaustive coverage of the surveillance region is unrealistic. An example use case is sea ice monitoring using Unmanned Aerial Systems (UASs) in the Arctic — a scenario which was used to exemplify the algorithm through two separate experimental datasets.

The proposed algorithm internally uses a Probability Hypothesis Density (PHD) representation, found in modern Multiple Target Tracking (MTT) methods, of unobserved targets. Approximations have been derived that allows for the PHD to be sequentially updated per agent, taking into consideration probabilities of targets being observed by previous agents as well as the probability of detection of each agent's sensor. This gives an algorithm that scales well with the number of agents used. An outline of the implementation has been presented in the article, and reference given to a Free and Open Source Software (FOSS) implementation available online.

The IPP algorithm has been evaluated in simulations, where it is shown to outperform both a greedy base line implementation and a classic lawn mover search pattern, when it comes to the expected number of observed targets. Furthermore, the method is shown to produce reasonable search paths when applied to experimental data from two different measurement campaigns in the Arctic region outside of Svalbard.

References

Brekke E, Chitre M (2018) Relationship between Finite Set Statistics and the Multiple Hypothesis Tracker. IEEE Transactions on Aerospace and Electronic Systems 9251:1–15, DOI 10.1109/TAES.2018.2805178

Chekuri C, Pál M (2005) A recursive greedy algorithm for walks in directed graphs. In: Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS, Pittsburgh, PA, USA, vol 2005, pp 245–253, DOI 10.1109/SFCS.2005.9

Chen SG (2014) Reduced recursive inclusion-exclusion principle for the probability of union events. In: IEEE International Conference on Industrial Engineering and Engineering Management, Bandar Sunway, Malaysia, pp 5–7, DOI 10.1109/IEEM.2014.7058590

Dames P (2017) Distributed multi-target search and tracking using the PHD filter. In: 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), IEEE, DOI 10.1109/mrs.2017.8250924, URL `https://doi.org/10.1109/mrs.2017.8250924`

Dames P (2019) Distributed multi-target search and tracking using the phd filter. Autonomous Robots DOI 10.1007/s10514-019-09840-9, URL `https://doi.org/10.1007/s10514-019-09840-9`

Fiorelli E, Leonard NE, Bhatta P, Paley DA, Bachmayer R, Fratantoni DM (2006) Multi-AUV Control and Adaptive Sampling in Monterey Bay. IEEE Journal of Oceanic Engineering 31(4):935–948, DOI 10.1109/JOE.2006.880429

Galceran E, Carreras M (2013) A Survey on Coverage Path Planning for Robotics. Robotics and Autonomous Systems 61(12):1258–1276, DOI 10.1016/j.robot.2013.09.004

Gordon N, Salmond D, Smith A (1993) Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. IEE Proceedings F - Radar and Signal Processing 140(2):107, DOI 10.1049/ip-f-2.1993.0015, `9241F(E5)`

Granström K, Lundquist (2012) Extended Target Tracking Using PHD Filters. PhD thesis, Linköping University, Linköping

Hollinger GA, Sukhatme GS (2014) Sampling-based Robotic Information Gathering Algorithms. International Journal of Robotics Research 33(9):1271–1287, DOI 10.1177/0278364914533443

IAMSAR (2010) IAMSAR Manual: International Aeronautical and Maritime Search and Rescue Manual. IMO

Kalman RE (1960) A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82(1):35, DOI 10.1115/1.3662552, `NIHMS150003`

Krause A, Guestrin C (2011) Submodularity and its Applications in Optimized Information Gathering. ACM Transactions on Intelligent Systems and Technology 2(4):1–20, DOI 10.1145/1989734.1989736

Mahler RPS (2003) Multitarget Bayes Filtering via First-Order Multitarget Moments. IEEE Transactions on Aerospace and Electronic Systems 39(4):1152–1178, DOI 10.1109/TAES.2003.1261119

Mahler RPS (2006) A theory of PHD filters of higher order in target number. In: Proceedings of SPIE, Orlando, FL, USA, 6235, May 2006, pp 1–12, DOI 10.1117/12.667083

Mahler RPS (2007a) PHD Filters of Higher Order in Target Number. IEEE Transactions on Aerospace and Electronic Systems 43(4):1523–1543, DOI 10.1109/TAES.2007.4441756

Mahler RPS (2007b) Statistical Multisource-Multitarget Information Fusion. Artech House, Inc., Norwood, MA, USA

Mahler RPS, Zajic T (2001) Multitarget filtering using a multitarget first-order moment statistic. Proceedings of SPIE 4380(August 2001), DOI 10.1117/12.436947

Mitchell M (1996) An Introduction to Genetic Algorithms. Computers & Mathematics with Applications 32(6):133, DOI 10.1016/S0898-1221(96)90227-8

Olofsson J, Brekke E, Fossen TI, Johansen TA (2017a) Spatially indexed clustering for scalable tracking of remotely sensed drift ice. In: IEEE Aerospace Conference Proceedings, Big Sky, MT, USA

Olofsson J, Veibäck C, Hendeby G (2017b) Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter. In: 20th International Conference on Information Fusion (FUSION), Xi'an, China

Pasha A, Vo BN, Tuan HD, Mat WK (2006) Closed form PHD filtering for linear jump Markov models. In: International Conference on Information Fusion (FUSION), Florence, Italy, DOI 10.1109/ICIF.2006.301593

Reid DB (1979) An Algorithm for Tracking Multiple Targets. IEEE Transactions on Automatic Control 24(6):843–854, DOI 10.1109/CDC.1978.268125

Reuter S, Vo BT, Vo BN, Dietmayer K (2014) The Labeled Multi-Bernoulli Filter. IEEE Transactions on Signal Processing 62(12):3246–3260, DOI 10.1109/LSP.2015.2414274

Sidenbladh H (2003) Multi-target particle filtering for the probability hypothesis density. In: Proceedings of the 6th International Conference of Information Fusion, Queensland Australia, vol 2, pp 800–806, DOI 10.1109/ICIF.2003.177321, 0303018v1

Singh A, Krause A, Guestrin C, Kaiser WJ (2009) Efficient Informative Sensing using Multiple Robots. Journal of Artificial Intelligence Research 34:707–755, DOI 10.1613/jair.2674, 1401.3462

Skoglar P (2012) Tracking and Planning for Surveillance Applications. PhD thesis, Linköping University

Streit R (2017) How I learned to stop worrying about a thousand and one filters and love analytic combinatorics. In: IEEE Aerospace Conference Proceedings, Big Sky, MT, USA, DOI 10.1109/AERO.2017.7943644

Vo BN, Ma WK (2006) The Gaussian Mixture Probability Hypothesis Density Filter. IEEE Transactions on Signal Processing 54(11):4091–4104, DOI 10.1109/TSP.2006.881190

Vo BN, Vo BT, Phung D (2014) Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter. IEEE Transactions on Signal Processing 62(24):6554–6567, DOI 10.1109/TSP.2014.2364014, arXiv:1312.2372v1

Vo BT, Vo BN (2013) Labeled Random Finite Sets and Multi-Object Conjugate Priors. IEEE Transactions on Signal Processing 61(13):3460–3475, DOI 10.1109/TSP.2013.2259822, arXiv:1312.2372v1

Vo BT, Vo BN, Cantoni A (2006) The cardinalized probability hypothesis density filter for linear Gaussian multi-target models. In: 2006 40th Annual Conference on Information Sciences and Systems, Princeton, New Jersey, USA, pp 681–686, DOI 10.1109/CISS.2006.286554

Vo BT, Vo BN, Cantoni A (2007) On multi-Bernoulli approximations to the Bayes multi-target filter. In: Proceedings of the 10th International Conference on Information Fusion, FUSION 2007, Québec, Canada

Vo BT, Vo BN, Cantoni A (2009) The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations. IEEE Transactions on Signal Processing 57(2):409–423, DOI 10.1109/TSP.2008.2007924

Wiener N (1965) Cybernetics, Second Edition: Or the Control and Communication in the Animal and the Machine. The MIT Press

Williams JL (2015) Marginal Multi-Bernoulli Filters: RFS Derivation of MHT, JIPDA, and Association-based MeMBer. IEEE Transactions on Aerospace and Electronic Systems 51(3):1664–1687, DOI 10.1109/TAES.2015.130550, 1203.2995

Winter CL, Stein MC (1993) An Additive Theory of Probabilistic Evidence Accrual. Advances in Applied Mathematics

Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, DOI 10.1109/ICPR.2004.1333992