

A GM-PHD Filter with Estimation of Probability of Detection and Survival for Individual Targets

R.A. Thivanka Perera¹, Mingi Jeong², Alberto Quattrini Li² and Paolo Stegagno¹

Abstract—This paper proposes a modification of the Gaussian mixture probability hypothesis density (GM-PHD) filter to compute online the probability of detection (P_D) and probability of survival (P_S) of targets. This eliminates the need for predetermined and/or constant P_D and P_S values, that may degrade the estimation. The proposed filter estimates the P_D and P_S values for each individual target based on newly introduced parameters, which are updated during the measurement update process. The effectiveness of the proposed filter was validated through an in-lab experiment using four unmanned ground robots with varying P_D values and a real-world lidar-based obstacle tracking system implemented on an Automated Surface Vehicle operating in a lake with real-time boat traffic. The results of the experiments demonstrate that the proposed filter outperforms the standard PHD filter with incorrect P_D and P_S values. These findings highlight the potential benefits of the proposed filter in improving target tracking performance in complex environments.

I. INTRODUCTION AND RELATED WORKS

This paper presents a multi-target tracking filter for the detection of obstacles on an Autonomous Surface Vehicle (ASV) within the context of a larger project for the development of robotic tools for cyanobacteria monitoring in ponds and lakes. In general, identification and tracking of obstacles is one of the basic tasks for reliable autonomous navigation. Often, obstacle avoidance algorithms are fed directly with sensor readings, but sensor noise, missed detections, false positives, and limited field of view can degrade the performance of the system and compromise safety. Proper Multi-Target Tracking (MTT) techniques [1] can be used to avoid or mitigate these issues. The most popular approaches in MTT are joint probabilistic data association filters [2], multiple hypothesis tracking [3], and random finite set (RFS) [4], the main difference being how data association is handled and whether the number of tracked targets is known or estimated.

The Probability Hypothesis Density (PHD) filter [4] belongs to the RFS category, does not require explicit data association, nor fixed or known number of targets. Therefore it is possible to employ it in generic unknown environments for obstacle avoidance purposes. It was originally developed to take into account many sensor unidealities by modeling the statistical properties of the occurrence of measurements and target survivability through appropriate parameters. The

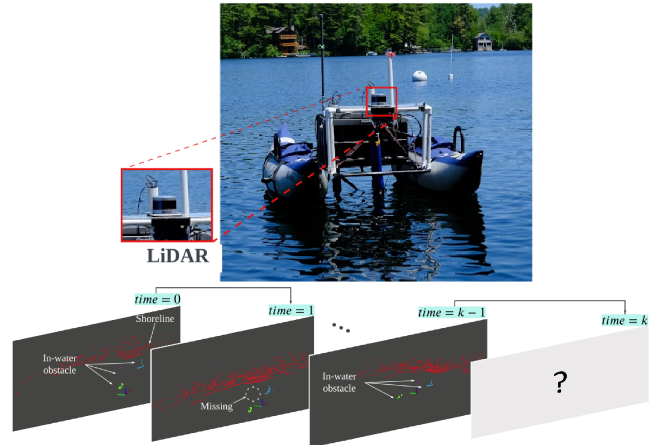


Fig. 1: Robotic boat Catabot with Velodyne VLP-16 scanner (top), measurements for the obstacle tracking, each obstacle with a different color (bottom).

Probability of Detection (P_D) and Probability of Survival (P_S) are two of such parameters, and their values affect the performance of the filter. In general, P_D indicates the confidence that a tracking filter has in the ability of the sensor to detect the target, while P_S describes the probability that a target disappears during a time update step. Incorrect values for P_D and P_S can lead to poor and unreliable tracking results, with the appearance of clutter in the estimates or the disappearance of targets. Multiple factors can affect the P_D and P_S of a target, including its state, clutter, sensor flaws, non-uniformity, and observation environment. Therefore, computing a global value for the P_D and P_S requires a complex combination that should take into account all of these factors, most of which are target dependent and unknown in real world. In the original formulation of the PHD filter however, P_D and P_S are described through state dependent functions that are the same for all targets.

In the context of ASV's (Fig.e 1) the identification of global P_D and P_S values associated with the sensor is made even more difficult by the wide variety of obstacles that can be encountered, including vessels, buoys, swimmers, and birds among others. Additionally, the attitude of an ASV constantly changes due to waves and wind disturbing the sensor readings that becomes less reliable for obstacles that are small, distant, and/or less elevated on the water surface [5], [6]. To address these issues, in this paper we follow a data-driven approach that, together with each target state, computes an individual estimate of P_D based on the history of the measurements associated to that target.

In literature, there has been a significant body of research

¹R.A. Thivanka Perera and Paolo Stegagno are with the Electrical, Computer and Biomedical Engineering Department, University of Rhode Island, 2 E Alumni Rd, Kingston, RI, USA thiva@uri.edu, pstegagno@uri.edu

²Mingi Jeong and Alberto Quattrini Li are with the Department of Computer Science, Dartmouth College, Hanover, NH USA mingi.jeong.gr@dartmouth.edu, alberto.quattrini.li@dartmouth.edu

focused on the estimation of the P_D . In [7], a Bayesian estimation method for P_D is introduced, although the study is limited to single target tracking instances. The most common method for estimating P_D is to relate it to the target state. In [8], predetermined P_D values are assigned to different parts of the sensor field, and the P_D for each estimate is derived based on where it is located in the sensor field. In [9], the authors propose a feature-based Inverse Gamma Gaussian Mixture (IGGM) distribution for P_D tracking in PHD and CPHD filters, where the feature is a part of the incoming measurements and is independent from the tracked state. In [10], the authors create a list that contains estimates with lower weights below the pruning threshold and introduced separate variables to track if subsequent recursion will receive any measurement to improve the associated weight, but this study does not address the core problem of variable P_D rates. In a study conducted by [11], individual P_D was investigated, in which the P_D was included in the state space and tracked using a separate distribution. However, the authors note that their approach is only effective for high P_D rates and comes with increased computational complexity.

In this paper we propose a modification of the GM-PHD filter where each target's P_D is individually estimated at run time, and P_S is computed to reflect such an estimate. The filter is tested on lab data in a multi-robot setup and on field data collected with an Autonomous Surface Vehicle (ASV). The rest of the paper is organized as follows. In Section II we provide the necessary background on the GM-PHD filter. In Section III we provide a description of the methodology to estimate the P_D and P_S . Section IV presents the experiments, while section V concludes the paper.

II. BACKGROUND

The background for the PHD filter and its equations are based on [4], [12]. We assume that at time step k , there are m_k targets living in the state space $\mathcal{R}^n$ with states $x_{1,k}, x_{2,k}, \dots, x_{m,k} \in \mathcal{X}$ where $\mathcal{X} \subseteq \mathcal{R}^n$. A measurement set $z_{1,k}, z_{2,k}, \dots, z_{m,k} \in \mathcal{Z}_k$ is obtained at time k which represents a finite subset from an observation space $\mathcal{Z} \in \mathcal{R}^m$.

The goal of the Bayesian multi-target recursive filter is to estimate the multi-target posterior $f_{k|k}(X_k|Z_{1:k})$ given a set of observations up to time step k over the set $\mathcal{X}_k$. Due to implementation limitations of the Bayesian multi-target filter, Mahler [4] proposed the first moments propagation of the multi-target posterior $f_{k|k}(X_k|Z_{1:k})$, called PHD filter. The PHD filter estimates the PHD of targets in $\mathcal{X}$ at time step k where it is defined as an intensity function $f_k(x)$ such that its integral over a subset $S \subseteq \mathcal{X}$ will provide the expected number of targets $N(S)$ in S , i.e., $N(S) = \int_S f_k(x)dx$. The PHD filter consists of two steps.

The prediction (time update) computes the PHD $f_{k|k-1}(x)$ at step k given all the measurements up to step $k-1$ from the PHD $f_{k-1|k-1}(x)$ at step $k-1$ given all the measurements up to step $k-1$ (i.e., predicting over one time step):

$$f_{k|k-1}(x) = \gamma_k(x) + \int [P_S(x')f_{k|k-1}(x|x') + b_{k|k-1}(x|x')]f_{k-1|k-1}(x')dx', \quad (1)$$

where $\gamma_k(x)$ is the probability that a new target appears in x between times $k-1$ and k , $P_S(x')$ is the probability that a target in x' at time $k-1$ will survive into step k , $f_{k|k-1}(x|x')$ is the probability density that a target in x' at time $k-1$ moves to x at time k , and $b_{k|k-1}(x|x')$ is the probability that a new target spawns in x at time k from a target in x' at time $k-1$.

The correction (measurement update) computes the PHD $f_{k|k}(x)$ at step k given all the measurements up to step k from the PHD $f_{k|k-1}(x)$ at step k given all the measurements up to $k-1$ (i.e., incorporating the measurement at time k):

$$f_{k|k}(x) = f_{k|k-1}(x) [1 - P_D(x) + \sum_{z \in \mathcal{Z}_k} \frac{P_D(x)g(z|x)}{\lambda c(z) + \int P_D(x')g(z|x')f_{k|k-1}(x')dx'}], \quad (2)$$

where $P_D(x)$ is the probability that a measurement is collected from a target with state x , $g(z|x)$ is the sensor likelihood function, and $\lambda c(z)$ expresses the probability that a given measurement z is a false positive.

The Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter [12] and the Sequential Monte Carlo PHD (SMC-PHD) filter [13], [14] are some of the variants of the PHD filter that can be implemented in real-world systems. In this study, we will focus on the GM-PHD filter, which approximates all PHD's in Equations (1) and (2) as sum of weighted Gaussian functions:

$$f_{*|*}(x) = \sum_i f_{*|*}^i(x) = \sum_i w_{*|*}^i \mathcal{N}(x; m_{*|*}^i, p_{*|*}^i), \quad (3)$$

where the symbol $w_{*|*}^i \mathcal{N}(x; m_{*|*}^i, p_{*|*}^i)$ represents a Gaussian function with weight $w_{*|*}^i$, mean $m_{*|*}^i$, and covariance $p_{*|*}^i$. The time update in the GM-PHD filter becomes:

$$f_{k|k-1} = b_{k|k-1}(x) + \sum_i w_{k-1|k-1}^i P_{S_k}^i \int f_{k|k-1}(x|x') f_{k-1|k-1}^i(x') dx'. \quad (4)$$

The measurement update in the GM-PHD filter becomes:

$$f_{k|k}(x) = \sum_i f_{k|k-1}^i(x) (1 - P_{D_k}^i) + \sum_i \sum_{z \in \mathcal{Z}_k} \frac{P_{D_k}^i f_{k|k-1}^i(x) g(z|x)}{\sum_i \int P_{D_k}^i g(z|x') f_{k|k-1}^i(x') dx'}. \quad (5)$$

At the end of the measurement update, a pruning and merging step is executed to remove components whose weight is lower than a given threshold, and to merge components that are close to each other. This is done to avoid exponential computational complexity.

III. FORMULATION OF THE PARAMETERS ESTIMATION

The time and measurement update of the mean $m_{*|*}^i$, covariance $p_{*|*}^i$, and weight $w_{*|*}^i$ of the generic i -th component $f_{*|*}^i(x)$ of the PHD follow the equations first presented in [15], and are omitted here for brevity. In this section, we focus on the modifications that are presented in this paper. We refer the reader to [15] for the details about the system and measurement models. This is also consistent with the

fact that the modification presented here is compatible with any generic GM-PHD filter.

A. Introduction of new parameters

From the original formulation of the PHD filter, the probability of detection $P_{D_k}^i$ and the survival probability $P_{S_k}^i$ of each component in Equations (4)-(5) are computed from $P_D(x)$ as $P_{D_k}^i = P_D(m_{k|k}^i)$ and $P_S(x)$ as $P_{S_k}^i = P_S(m_{k|k}^i)$ respectively. Therefore, $P_{D_k}^i$ and $P_{S_k}^i$ depend only from the state of each component due to the sensor characteristics, and not from the target characteristics. However, in some instances where the probability of detection is governed by individual target characteristics, this assumption may not hold. In this proposed method, three additional parameters are introduced to the usual state of a target. We begin with Equation (3), where we introduce the detection counter α_k^i , the missed detection counter β_k^i and the number of missed detection since last detected λ_k^i . As their names imply, α_k^i represents the number of times the target represented by the component $f_{k|k}^i(x)$ has been detected since birth up to time step k , β_k^i represents the number of times the target has been missed since birth up to time step k and λ_k^i represents the number of times that this target has not been detected since last time it was detected. Therefore, we define the expanded component $\phi_{k|*}^i(x)$ as the tuple

$$\begin{aligned}\phi_{k|k-1}^i(x) &= \{f_{k|k-1}^i(x), \alpha_{k-1}^i, \beta_{k-1}^i, \lambda_{k-1}^i\} \\ \phi_{k|k}^i(x) &= \{f_{k|k}^i(x), \alpha_k^i, \beta_k^i, \lambda_k^i\}.\end{aligned}\quad (6)$$

Note that the values of α_k^i , β_k^i , λ_k^i will not change during the time update step, but only during the measurement update.

B. Online P_D Estimation

The probability of detection for the time step k will be calculated based on each estimate's $\alpha_{k-1}^i, \beta_{k-1}^i$ values as:

$$P_{D_k}^i = \frac{\alpha_{k-1}^i}{\beta_{k-1}^i + \alpha_{k-1}^i}, \quad (7)$$

where $\beta_{k-1}^i + \alpha_{k-1}^i$ is the total number of measurement updates since the birth of component i up to time step $k-1$.

C. Online P_S Calculation

In general, the probability of survival can be used as a mechanism to remove estimates of targets that have likely disappeared, based on the consecutive times that a target has not been detected. However, targets with lower probability of detection P_D^i will experience longer streaks of consecutive missed detections, as the probability of m consecutive misses is $(1 - P_D^i)^m$. Therefore, we have implemented the P_S of each target as a function of the probability of detection $P_{D_k}^i$ and the steps without measurements λ_{k-1}^i as:

$$P_{S_k}^i = \frac{1}{1.1 + 0.5\lambda_{k-1}^i P_{D_k}^i} \cdot \frac{1}{1 + \exp(3(\lambda_{k-1}^i - C))}. \quad (8)$$

The following assumptions were made in developing Equation (8). It is assumed that a component with higher P_D value have a positive correlation with P_S and an inverse

correlation with λ^i . This assumption is reasonable in a general tracking environment. For instance, if a higher P_D target is missed by the sensor for several time steps, it indicates that the target is no longer viable and, thus $P_{S_k}^i$ should decrease at a higher rate for every increment of λ^i value. Conversely, lower P_D components are expected to be missed by the sensor for several time steps, and thus they should persist into the next time step with higher P_S values. This behavior is captured in the first factor of Equation (8). The second factor is a threshold to limit the maximum number of missed measurements that are allowed for any target, independently from the estimated probability of detection. In fact, $\frac{1}{1 + \exp(3(\lambda_{k-1}^i - C))}$ is a sigmoid function that is 1 for $\lambda_{k-1}^i < C$, and 0 for $\lambda_{k-1}^i > C$. The variable C represents the maximum number of allowed consecutive missed detections before a component is eliminated.

This structure for the $P_{S_k}^i$ is needed to avoid a situation in which a target's estimated $P_{D_k}^i$ tends to 0, and therefore the component is never eliminated from the filter even though it is never associated to a measurement. In practice, this could more likely happen in environments with significant clutter, therefore, when false positive measurements are expected to be collected at higher rates, the parameter C should be lower with respect to environments in which false positive measurements are not expected.

D. Updating scheme for new parameters

The update of α_k^i , β_k^i , λ_k^i will be done only during the measurement update in Equation (5) which includes two terms. The first sum in the right hand side includes the components originating from missed detections, i.e., it includes all components that are not associated to any measurement at the time step k . These components are exact copies of the components of $f_{k|k-1}$ with weights multiplied by $(1 - P_{D_k}^i)$. Therefore, for these components we update the counters as:

$$\alpha_k^i = \alpha_{k-1}^i, \quad \beta_k^i = \beta_{k-1}^i + 1, \quad \lambda_k^i = \lambda_{k-1}^i + 1. \quad (9)$$

The α_k^i will simply propagate the previous time step value, while the counters β_k^i and λ_k^i will be increased by one. In the same Equation (5), the second sum in the right hand side represents all the components that originates from associating a measurement to all the components of the prior $f_{k|k-1}$. For these components, α_k^i will be incremented by one to represent the measurement association, λ_k^i will be reset to zero which indicates that each of these components was associated with a measurement, while β_k^i will not be updated. The update equations are:

$$\alpha_k^i = \alpha_{k-1}^i + 1, \quad \beta_k^i = \beta_{k-1}^i, \quad \lambda_k^i = 0. \quad (10)$$

E. Merging and Pruning

During the measurement update stage of the GM-PHD filter, $|Z_k| + 1$ components are generated for each $f_{k|k-1}^i(x)$ component of the prior through equations (5), (9) and (10), where the symbol $|*|$ indicates the cardinality of a set. To avoid exponential computational complexity, a

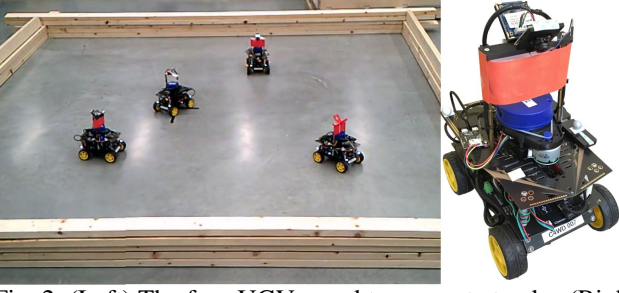


Fig. 2: (Left) The four UGVs used to generate tracks. (Right) A Single UGV [15].

merging and pruning step is executed at the end of the measurement update. In general, in the merging step, the components $f_{k|k}^i(x)$ are partitioned into clusters of closely spaced Gaussian components. We indicate with I_k^h the set of all indexes of the components belonging to the h -th cluster. Therefore, the set of all components in the h -th cluster is $F^h = \{\phi_{k|k}^i(x) : i \in I_k^h\}$. These will be replaced with a single component $\phi_{k|k}^H(x)$ such that:

$$\begin{aligned} w_{k|k}^H &= \sum_{i \in I_k^h} w_{k|k}^i, & m_{k|k}^H &= \frac{1}{w_{k|k}^H} \sum_{i \in I_k^h} w_{k|k}^i m_{k|k}^i \\ p_{k|k}^H &= \frac{1}{w_{k|k}^H} \sum_{i \in I_k^h} w_{k|k}^i (p_{k|k}^i + (m_{k|k}^H - m_{k|k}^i)(m_{k|k}^H - m_{k|k}^i)^T) \\ \alpha_k^H &= \alpha_k^j, \beta_k^H = \beta_k^j, \lambda_k^H = \lambda_k^j, & \text{where } j &= \arg \max_{i \in I_k^h} w_{k|k}^i. \end{aligned} \quad (11)$$

Note that the merging of the components follows the standard merging of the GM-PHD filter proposed in [12]. For the additional variables $\alpha_k^H, \beta_k^H, \lambda_k^H$, we use the value of the component with the highest weight. Other merging procedures were tested during the development of the filter, particularly the weighted mean. However, it is possible to prove that such choice would result in underestimating the probability of detection, due to the overestimation of the parameter β_k^H . This happens because a missed measurement generates only a component with increased β_k^H , while a collected measurement generates two components, one with increased β_k^H and one with increased α_k^H .

The estimation of the P_D can be performed with infinite memory to obtain more stable estimates, or with limited memory considering only the latest N steps, to obtain a more reactive system. The latter approach is useful in case of obstacles with variable P_D . Further, an initial value greater than zero must be assigned to α^i, β^i at initialization to avoid an indeterminate value for P_D (Equation (7)).

IV. EXPERIMENTS

A. In-lab Experiment

Experiments were conducted in laboratory to test the proposed filter using four unmanned ground vehicles (UGVs) (Fig. 2), which allowed us to evaluate the filter performance with ground truth from a motion capture system. The UGVs were built using a commercially available differential drive platform, the DFRobot Cherokee (22.5cm x 17.5cm). Each UGV was equipped with wheel encoders, an Arduino Romeo V2 with a motor driver for low level control, and an

Odroid-XU4 to manage sensor data collection. In a typical experiment, the UGVs perform a pseudo-random motion with obstacle avoidance within a testing area equipped with an Opti-track motion capture system. Odometry and tracking information was recorded to a ROS bag.

The robot locations collected by the Opti-track system were used to simulate measurements and as ground truth. In the experiment presented in this section, the four robots were assigned a true probability of detection of 0.3, 0.5, 0.7, 0.9, respectively. At this aim, a separate algorithm extracted each robot's position from the recorded ROS bag and provide it to the filter based on the assigned P_D . For example, a robot assigned with 0.7 P_D will provide its measurement to the filter 7 out of 10 time steps. Further, a Gaussian zero mean white noise with standard deviation 0.02m was added to each robot coordinate to simulate sensor noise. As the filter estimated the position of the robots in the Opti-track fixed frame of reference, the ego-motion of the estimator required in [15] is always selected as zero.

To highlight the benefits of the proposed filter, we simultaneously ran a standard PHD filter with a predetermined constant P_D value, and our modified PHD filter with online estimation of the P_D . We processed the recorded dataset three times using the standard PHD filter with constant P_D values of 0.9, 0.5, 0.2 as shown in Fig. 3(a), (b) and (c), respectively. Note that the results of different runs of the adaptive PHD filter present slight variations because, while the ground truth data is the same for each run from the ROS bag, the simulated measurement were created at run time, and therefore have different noise samples and different sequences of collected and missed measurements. The values for the P_D were selected to represent a good estimate of the actual P_D (0.5), and two situations in which the P_D was either overestimated (0.9) or underestimated (0.2). The variable C was selected as 20 and each estimate was initialized with $\alpha^i = 3, \beta^i = 1$, which result in an initial $P_D = 0.75$. We generated four different plots: the moving average of the sum of estimate's weights for each time steps (Fig. 3 Row 1), the estimated values for P_D^i (Fig. 3 Row 2), the XY Cartesian plot with the estimates and the ground truth positions (Fig. 3 Row 3), and the moving average of the estimation error (Fig. 3 Row 4). In Fig. 3, the standard filter results are plotted in red, while the adaptive PHD filter results are plotted in green. In the XY Cartesian plot, the ground truth is plotted in yellow.

The proposed filter was able to accurately estimate the P_D of three robots in all three instances. The lowest P_D 0.3 was estimated during several time steps but overall its P_D was overestimated. This was expected as the estimates with lowest P_D were often removed from the filter during the pruning, and the relative component were often reinitialized. Overall, the adaptive PHD filter was able to track four robots as seen from the overall sum of weights (Fig. 3 Row 1), whereas the number of robots estimated by the standard PHD filter depends on the selected probability of detection.

The Cartesian plot (Row 4) shows that the robots were properly tracked by the proposed filter, whereas the quality

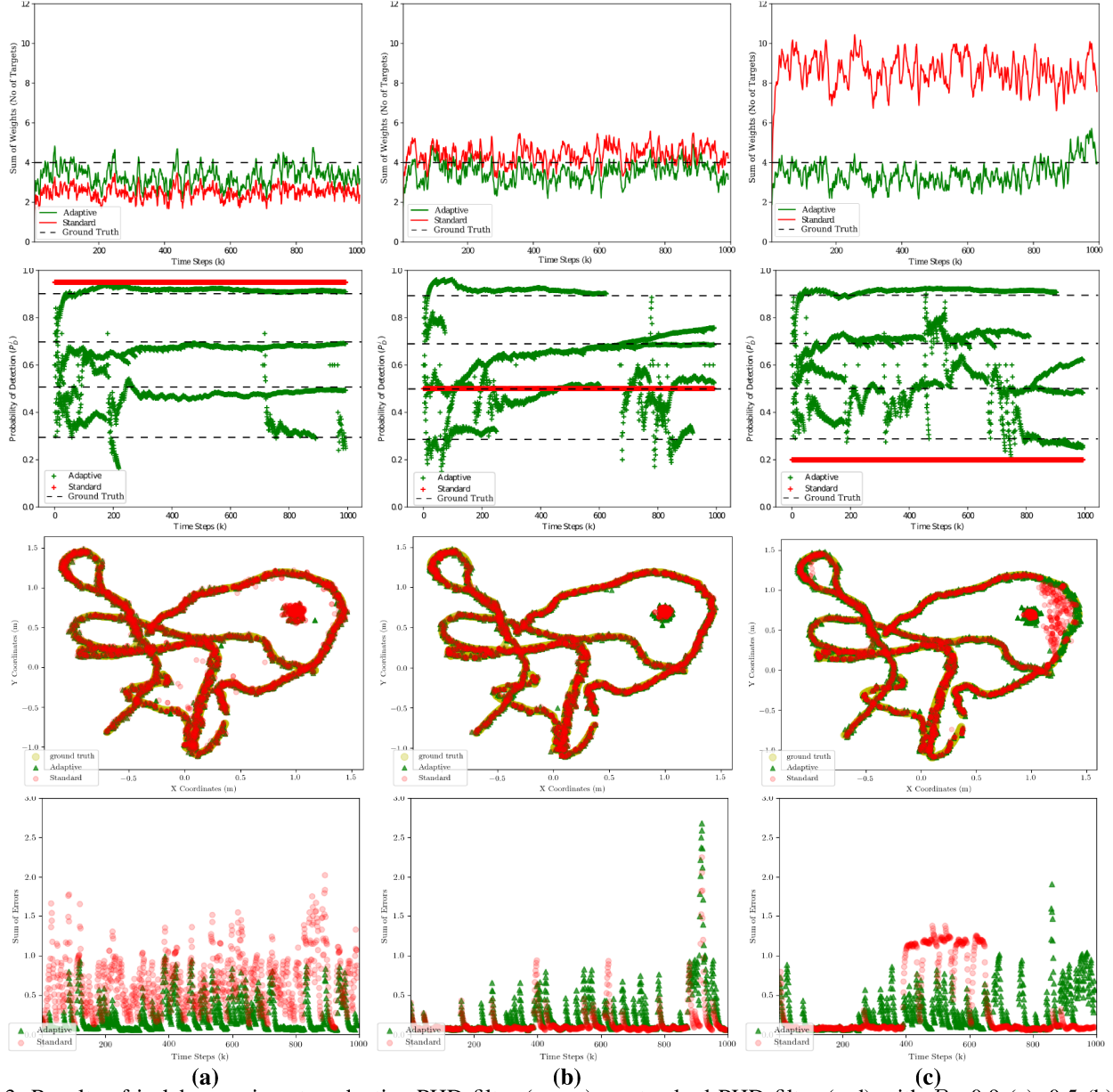


Fig. 3: Results of in-lab experiments: adaptive PHD filter (green) vs standard PHD filter (red) with P_D 0.9 (a), 0.5 (b) and 0.2 (c). Row 1: moving average of sums of weights for each time step; row 2: P_D values of every estimate; row 3: XY Cartesian plot with estimates and ground truth; row 4: sum of estimation errors.

of the tracking in the standard PHD filter depends on the selected P_D value. In particular, the filters with P_D 0.9 and 0.2 show considerable amount of estimation error along the robot's paths. The standard PHD filter with $P_D = 0.2$ exhibited significant error between the x-coordinates 0.5-1.5m and y-coordinates 0-1m, where a moving robot passed by a stationary robot. This error was attributed to the standard PHD filter mistakenly estimating one robot instead of two at this time, leading to larger tracking errors. Overall these comparisons clearly shows the advantage of proposed filter.

This is further corroborated by the error plot depicted in Fig. 3 Row 5. The overall error was computed by summing the individual robot estimation error. The estimates were assigned to the closest robot through a clustering algorithm, ensuring that each estimate was only associated with a single

robot. In the case where an estimate was not available for a particular robot, a maximum error of 1 meter was assigned, so that a maximum error of 4m is possible. The error plots of the standard PHD filter with P_D 0.2 and 0.9 are outperformed by the adaptive PHD filter, which has 2.76 and 1.41 times less error respectively. However, the standard PHD filter with $P_D = 0.5$ shows better performance with respect to our proposed filter, due to the more frequent re-initializations of one of the estimates. This behavior is expected as the average actual P_D of the four robots is 0.6, which makes a constant overall P_D of 0.5 a good estimate. The mean error values are proposed in Table I.

Figure 4 displays the results of another experiment in which the P_D of the four robot is reduced over time, from 1, 0.9, 0.8, 0.7, at the beginning of the experiment to 0.75,

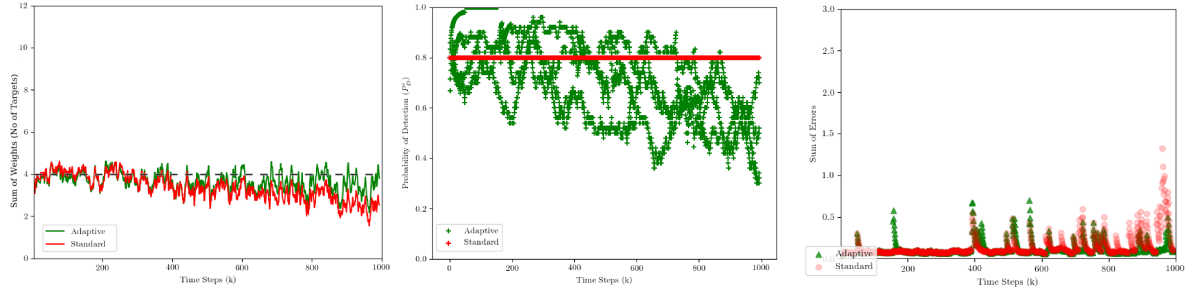


Fig. 4: Results of in-lab experiment with variable target P_D : adaptive PHD filter (green) vs standard PHD filter (red). Left: moving average of the sum of weights; middle: P_D values of every estimate; right: sum of the estimation errors.

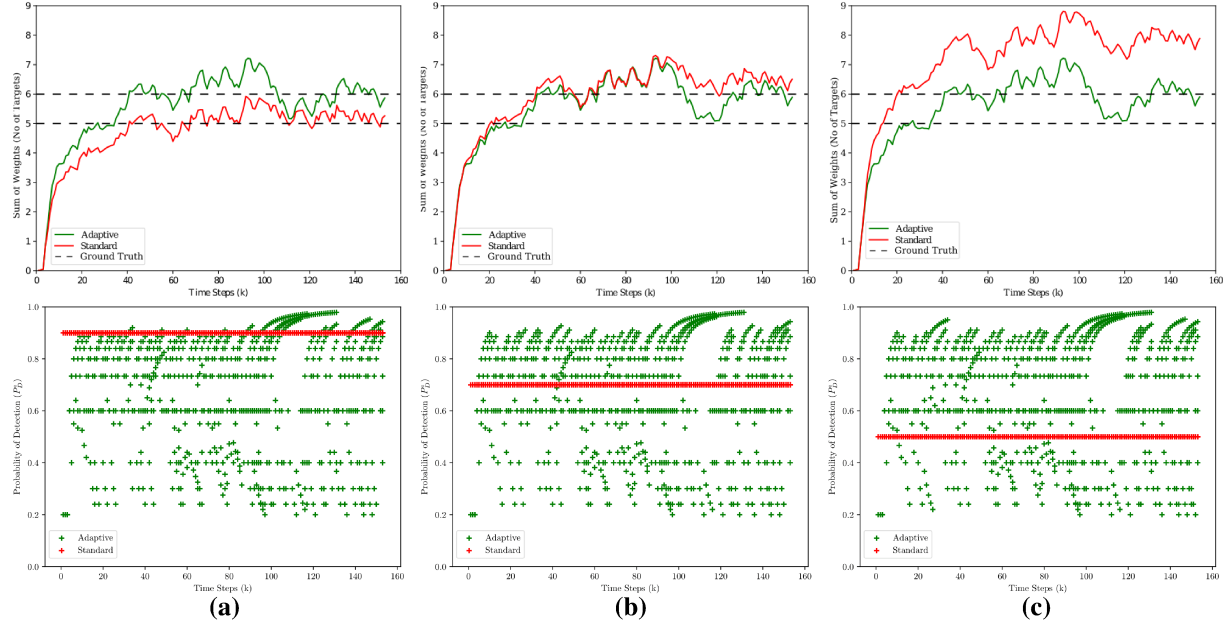


Fig. 5: Results of real-world ASV experiments: adaptive PHD filter (green) vs standard PHD filter (red) with P_D 0.9 (a), 0.7 (b) and 0.5 (c). Row 1: Moving average of sums of weights for each time step and, Row 2: P_D values of each estimate.

Filter Type	Standard	Adaptive	percentage
$P_D = 0.9$	0.737	0.267	2.76
$P_D = 0.5$	0.156	0.256	0.6
$P_D = 0.2$	0.337	0.239	1.41

TABLE I: Mean error values in the various experiments.

0.65, 0.55, and 0.45 at the end. The adaptive filter was run with a memory window for the P_D estimation of 50 steps, and was compared with the standard PHD filter with a fixed P_D of 0.8. The plots present the sum of weights, the estimated P_D , and the overall error, demonstrating that both filters have similar performance at the beginning, but as the P_D decreased the adaptive filter was able to maintain its performance, while the performance of the standard PHD filter deteriorated.

B. Real-world ASV Obstacle Tracking Experiment

We have tested our proposed filter on data collected by a small ASV in a lake environment, where P_D and P_S are unknown. We utilized our custom-designed 25 kg ASV Catabot (Fig. 1), measuring 2.4 m in length and 1.4 m in width [16]. The ASV is equipped with a Velodyne VLP-16 LiDAR for 3D measurements, offering a 360° horizontal and 30° vertical field of view, as well as other components like an

Intel NUC onboard computer, an RGB camera, GPS, IMU, and sonar. The LiDAR was installed at the center of the ASV minimizing occlusion caused by other fixed components. Note that the proposed tracking method can function with any LiDAR sensor that can generate a point cloud with x, y, and z geometric data.

We collected a vast array of data during the ice-out season from Apr. 2020 to Sep. 2022 in Lake Sunapee in NH and Lake China, Sabattus, and Auburn in ME. The objects we encountered during this time were diverse, ranging from swimmers and various types of buoys to power boats, water skiers, kayaks, floating docks, and sailboats with encounters in different situations (such as head-on, crossing, overtaking, and overtaken) and various ASV ego-motions (such as stopping, translating, turning). Targets tracked in this experiment are from real lake traffic on which we did not have control. Therefore, no ground truth is available to compute the estimation error, and the best indicator for success is the estimated number of targets.

To ensure that our proposed tracking approach is effective in real-world scenarios, we employed a model-free segmentation algorithm that we had developed in our previous work [6] as a pre-processing step. To minimize computational

requirements, our method converts the 3D point cloud into a 2D spherical projection image. The algorithm used for segmentation integrates a breadth-first search with a variant of hierarchical agglomerative clustering to segment the points based on the different objects present in the environment. This approach addresses the sparsity of the point cloud in the aquatic domain, which is a characteristic that makes methods developed for self-driving cars unsuitable for in-water obstacle segmentation. Once the segmentation is complete, our algorithm computes the centroids of the segmented output and provides it to the PHD filter for the measurement update. Ego-motion measurements of the Catabot for the time update were provided directly from the onboard localization module.

Similar to in-lab experiments we simultaneously ran the standard PHD filter with different predetermined P_D (0.9, 0.7, 0.5) and P_S (0.9, 0.9, 0.8) values along with our adaptive PHD filter in a single sequence. The variable C was selected as 5. Each estimate was initialized with $P_D = 0.75$ by assigning initial value as $\alpha^i = 3$, $\beta^i = 1$. The results of the three executions are presented in Fig. 5 with two plots: the moving average of the sum of the weights (Fig. 5 Row 1), and the estimated values for P_D^i (Fig. 5 Row 2). In average, the lidar tracked 5 to 6 targets including a sail boat, a medium motor boat, kayak and several other objects as counted from the recorded video and lidar blobs during the experiment. As seen in Fig. 5 Row 1, our filter was able to track an average of 6 targets. The number of targets tracked by the standard PHD filter instead depends on the P_D , with the filters with P_D 0.9 and 0.5 (columns a and c respectively) underestimating and overestimating the number of targets, and the filter with P_D 0.7 showing similar results to our filter. The estimated P_D shows a few tracks with high values (0.8 and up) in the proposed adaptive filter in Fig. 5 Row 2 which could be explained by the sail boat and the medium size motor boat, that due to their larger sized produced more stable point clusters in the pointcloud.

Overall, these results corroborates the fact that manually selecting values for P_D and P_S in the standard PHD filter will change the results of the estimation, while using an estimation method to compute individualized P_D and P_S values for each target produces more consistent results.

V. CONCLUSIONS

This paper introduces a modification of the GM-PHD filter with adaptive probability of detection (P_D) and probability of survival (P_S), thereby eliminating the need for predetermined and constant P_D and P_S values. The proposed filter estimates the P_D and P_S values for each individual target based on newly introduced parameters, which are updated during the measurement update process. We performed in-lab experiments using four unmanned ground robots, each with a unique P_D value, and compared against the standard PHD filter with several fixed P_D values. The results shows that our modified filter outperforms the standard PHD filter when the selected P_D and P_S are not accurate. This is evident both in the number of the estimated targets, as well as in the tracking error which is greatly reduced. On the other hand,

accurate values of P_D and P_S in the standard PHD filter may lead to comparable or slightly better results. These findings corroborate the fact that our system is suitable and provide an advantage when the values for P_D and P_S are unknown and should be guessed and/or tuned, as is for example the case of obstacle tracking on small ASV vehicles. Following this analysis, the proposed filter was applied to a dataset collected by an ASV operating in a real-world lake environment with boat traffic showing promising results.

In the future, we plan to extend the system to multi-sensor setup and perform live testing of the estimation system with live boat traffic and automated obstacle avoidance.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant RII Track-2 FEC 1923004 and CNS-1919647.

REFERENCES

- [1] B.-n. Vo, M. Mallick, Y. Bar-shalom, S. Coraluppi, R. Osborne III, R. Mahler, and B.-t. Vo, *Multitarget Tracking*. John Wiley and Sons, Ltd, 2015, pp. 1–15.
- [2] P. W. Y. Bar-Shalom and X. Tian, *Tracking and data fusion: Handbook of Algorithms*. YBS Publishing, 2012.
- [3] R. F. P. Samuel S. Blackman, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [4] R. Mahler, “Multitarget bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [5] R. Halterman and M. Bruch, “Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection,” in *Unmanned Systems Technology XII*, vol. 7692. SPIE, 2010, pp. 123–130.
- [6] M. Jeong and A. Q. Li, “Efficient lidar-based in-water obstacle detection and segmentation by autonomous surface vehicles in aquatic environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5387–5394.
- [7] K. G. Jamieson, M. R. Gupta, and D. W. Krout, “Sequential bayesian estimation of the probability of detection for tracking,” in *2009 12th International Conference on Information Fusion*, 2009, pp. 641–648.
- [8] G. Hendeby and R. Karlsson, “Gaussian mixture phd filtering with variable probability of detection,” in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–7.
- [9] C. Li, W. Wang, T. Kirubarajan, J. Sun, and P. Lei, “Phd and cphd filtering with unknown detection probability,” *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3784–3798, 2018.
- [10] M. Yazdian-Dehkordi and Z. Azimifar, “Refined gm-phd tracker for tracking targets in possible subsequent missed detections,” *Signal Processing*, vol. 116, pp. 112–126, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168415001437>
- [11] R. P. S. Mahler, B.-T. Vo, and B.-N. Vo, “Cphd filtering with unknown clutter rate and detection profile,” *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3497–3513, 2011.
- [12] B. . Vo and W. . Ma, “The gaussian mixture probability hypothesis density filter,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.
- [13] T. Zajic and R. P. S. Mahler, “Particle-systems implementation of the PHD multitarget tracking filter,” in *Signal Processing, Sensor Fusion, and Target Recognition XII*, I. Kadar, Ed., vol. 5096. SPIE, 2003, pp. 291 – 299.
- [14] B.-N. Vo, S. Singh, and A. Doucet, “Sequential monte carlo implementation of the phd filter for multi-target tracking,” vol. 2, 02 2003, pp. 792– 799.
- [15] R. A. Thivanka Perera, C. Yuan, and P. Stegagno, “A phd filter based localization system for robotic swarms,” in *Distributed Autonomous Robotic Systems*, F. Matsuno, S.-i. Azuma, and M. Yamamoto, Eds. Cham: Springer International Publishing, 2022, pp. 176–189.
- [16] M. Jeong, M. Roznere, S. Lensgraf, A. Sniffen, D. Balkcom, and A. Q. Li, “Catabot: Autonomous surface vehicle with an optimized design for environmental monitoring,” in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 2020, pp. 1–9.