Multi-Class Target Tracking Using the Semantic PHD Filter

Jun Chen and Philip Dames

Temple University, Philadelphia, PA 19122, USA,
{jchen,pdames}@temple.edu,
https://sites.temple.edu/trail/

Abstract. In order for a mobile robot to be able to effectively operate in complex, dynamic environments it must be capable of understanding both where and what the objects around them are. In this paper we introduce the semantic probability hypothesis density (SPHD) filter, which allows robots to simultaneously track multiple classes of targets despite measurement uncertainty, including false positive detections, false negative detections, measurement noise, and target misclassification. The SPHD filter is capable of incorporating a different motion model for each type of target and of functioning in situations where the number of targets is unknown and time-varying. We demonstrate the efficacy of the SPHD filter via simulations with multiple target types containing both static and dynamic targets. We show that the SPHD filter performs better than a collection of PHD filters running in parallel, one for each target class.

Keywords: AI-enabled Robotics, Robot Learning

1 Introduction

Multi-target tracking is a fundamental problem in robotics, wherein a robot simultaneously estimates the states of a potentially large number of individual objects. These objects can be either static or dynamic, and may leave or enter the area over time. In addition to tracking the kinematic or dynamic state of each target, as is standard, we believe that it is important for robots to be capable of tracking the semantic state, *i.e.,* what type of object is it? For example, if a household robot is tasked with setting the table, it must be able to recognize many different objects (*e.g.,* napkins, plates, forks, spoons, knives, cups, tables, drawers, and cabinets), the locations of each object, and the affordances of each object (*e.g.,* drawers slide open while cabinets swing open), which could be stored in a reference database. Achieving this requires a robot to be equipped with a sensor capable of measuring both position and the semantic label, such as an RGB-D camera. While real-time machine vision algorithms are reaching high levels of object classification accuracy [22–24, 32], there may still be significant uncertainty in the semantic label and it is often at the expense of high false positive rates.

Note in this paper we consider the task of mapping as a subset of multi-target tracking wherein all targets are stationary. Most existing approaches to robot localization and mapping collect low-level geometric features such as points, lines and planes [28]. However, by doing this robots only have the ability to plan paths and navigate through a geometrical map, which is different from apprehending the environment the way a human does and navigating from place to place [13]. Recently some work introduced methods of semantic localization [1] and semantic SLAM [2,3,5,12,25]. However, these works assume that all objects in the map are stationary, which is often not the case in complex, real-world environments.

Multi-target tracking (MTT) algorithms were originally developed to track dynamic objects. The MTT task is challenging due to the difficulty of solving the data association problem (*i.e.,* matching measurements to targets), which is further exacerbated by the possibility of false positive or false negative detections. Stone *et al.* [27] discuss in their book a number of probabilistic, multi-target tracking approaches, including the Multiple Hypothesis Tracker (MHT) [4], Joint Probabilistic Data Association (JPDA) [11], and the Probability Hypothesis Density (PHD) filter [17]. All of these approaches simultaneously solve the data association and tracking problems in different ways. The MHT makes hard associations and maintains a tree over the history of these associations. This results in unique tracks for each target. However, the number of branches in the tree grows quickly, requiring aggressive pruning algorithms that can lead to suboptimal performance. The JPDA makes soft associations and uses multiple measurements to update each target at each time step, which does not scale well with the number of targets. Finally, the PHD does not require any explicit choice for data association. As a result, the PHD does not actually distinguish between individual objects but rather represents the spatial density of objects. We argue that this is sufficient for a great amount of tasks, where it is only important to know what and where objects are but not to distinguish between objects of the same type. For example, navigating through an office environment does not require the robot to know *which* chair it is passing by, only that *a* chair is nearby. However, currently none of these existing trackers are able to utilize semantic measurements to track multiple types.

A number of methods have been provided for labeled tracking with the PHD filter, *i.e.,* uniquely identifying each individual object. Lin *et al.* [16] proposed a track labeling method by extracting peaks from the estimated PHD and correlating these over time. Vo *et al.* [30, 31] proposed a multi-target tracking filter using multi-object conjugate priors constructed by labeled RFSs, which is the first RFS-based multi-target filter that produces track-valued estimate in a principled manner. While these methods successfully solved the data association problem and thus realized multi-target labeled tracking, the huge computational load for each target to match its label may not be necessary in many scenarios where targets are classified by labels and labeling within a class is not important.

The PHD filter has been used in other contexts within robotics in the past. Mullane *et al.* [20] proposed an integrated Bayesian frame-work for SLAM in the general case of uncertain feature number and data association. This approach was then extended to track two types of objects, one static and one dynamic [15,19], however this lacks the ability to differentiate between different types of static and/or dynamic objects. Dames *et al.* [7] enabled a decentralized team of robot to autonomously explore an environment to detect and localize an unknown number of targets using the PHD filter. Dames [10] later introduced a distributed algorithm for multiple robots to search and track multiple targets in a coordinated manner using the PHD filter.

In this paper, we introduce the semantic probability hypothesis density (SPHD) filter, a modified version of the PHD filter that allows a robot to track any number of target classes. We demonstrate the efficacy of this method through a series of search and track tasks with multiple classes of either static or dynamic targets, and evaluate the results it achieved. The SPHD filter can be easily generalized to multiple robots using the approaches from the authors' past work [7,10].

2 The Semantic PHD (SPHD) Filter

A robot is tasked with exploring an environment $E \subset R^2$. Its pose at time t is $q^t \in SO(2)$. There are multiple targets of multiple types within the environment E and there may be multiple targets of each type, so not every target is uniquely identifiable. The number of targets of each class of targets is unknown and may change over time due to the motion of targets into and out of the environment. The key to defining the SPHD filter is to augment the dynamic state of a target with a discrete class label, *e.g.,* $x \in \mathcal{X} = R^2 \times C$, where $C = \{c_1, ..., c_k\}$ is a set of discrete class labels. We differentiate these two parts of the state space as the metric part $x^m \in \mathcal{X}^m$ (*e.g.,* $\mathcal{X}^m = R^2$) and the semantic part $x^s \in \mathcal{X}^s = C$.

At each time step, the robot collects a set of local measurements, $Z^t = \{z_1^t, \dots, z_m^t\}$. The number of measurements changes over time due to false positive and false negative detections as well as motion of the robot and targets causing target to enter and leave the sensor field of view (FoV). Each measurement z^t contains both metric information z^m (*e.g.,* range and bearing) and a detected class $z^s \in C$. The robot seeks to find all targets in the set $X^t = \{x_1^t, \dots, x_n^t\}$, where each $x_i^t \in E$. Note that this set encodes both the number of targets (*i.e.,* the cardinality of the set $|X^t|$) and the state of each target (*i.e.,* the elements of the set x_i^t), which includes both metric and semantic information.

2.1 Random Finite Sets

The sets X and Z from above are realizations of random finite sets (RFSs). An RFS is a set containing a random number of random elements, *e.g.,* each of the n elements x_i in the set $X = \{x_1, \dots, x_n\}$ is a vector indicating the state of a single target. See Mahler [18] for a more thorough treatment of the mathematics presented in this section. In deriving the PHD filter, Mahler [17] assumes that: 1) the clutter and true measurement RFSs are independent and 2) the clutter, target, and birth RFSs are Poisson. The first assumption is standard for target localization tasks. The second assumption is a result of assuming that the number of points in each finite region is independent if the regions do not overlap [6]. A Poisson RFS is one that has independently and identically distributed (i.i.d.) elements and where the number of elements follows a Poisson distribution. The likelihood of such an RFS X is

$$p(X) = e^{-\lambda} \prod_{x \in X} v(x), \tag{1}$$

where $v(\cdot)$ is the *Probability Hypothesis Density* (PHD), $\lambda = \int_E v(x)\,dx$, and $p(\varnothing) = e^{-\lambda}$. The PHD is a density function over the state space of the targets, with the unique property that the integral of the PHD over a region $S \subseteq E$ is the expected cardinality of an RFS X in that region. The PHD is also the first statistical moment of a distribution over RFSs. Note that it is *not* a probability density function, but it may be turned into one by normalizing by the expected cardinality,

$$p(x) = \lambda^{-1} v(x). \tag{2}$$

2.2 SPHD Models

The (S)PHD filter recursively updates the PHD using models of target motion and the measurement sets collected by the robots. Targets may move about within the environment, may appear in the environment, or may disappear from the environment. Each of these phenomena is explained by a target model. The **target motion model**, $f(x \mid \xi)$, describes the probability of a target transitioning from an initial state ξ to a new state x. While this may, in theory, allow targets to transition between different classes (*e.g.,* sitting person, standing person, and walking person could be different classes), we ignore this possibility in this paper. Instead, we assume there is a collection of class-dependent metric motion models, $f(x^m \mid \xi^m, \xi^s = c)$, $\forall c \in C$. The **birth model**, $b(x)$, is a PHD that describes both the number and states (including classes) of the new targets entering the environment. For many situations the birth PHD will only be nonzero near the boundaries of the environment, where new targets can enter the area of interest, and only for dynamic objects. Finally, the **survival probability**, $p_s(x)$, models the survival (and conversely the disappearance) of a target with state x. The birth and survival models also typically take the form of a collection of class dependent models, *i.e.,* the birth and survival process is different for each class type.

Each robot is equipped with a sensor to detect targets. This sensor may experience false negative detections, return noisy measurements to true targets, or receive false positive detections. Each of these phenomena is covered by a different sensor model. The **detection model**, $p_d(x \mid q)$, of a robot with state q detecting a target with state x characterizes the true (and false negative) detections. Note that the probability of detection is identically zero for all x outside the sensor field of view (FoV). In principal the detection likelihood could be different for each class, but in this paper we assume that it is independent of class, *i.e.,* $p_d(x \mid q) = p_d(x^m \mid q)$. The **observation model**, $g(z \mid x, q)$, returns a measurement z for a target with state x that is detected by a robot with state q. Like the target state space, the measurement also contains two separate parts: the metric part z^m and the semantic part $z^s \in C$. We assume that these two parts are independent conditioned on the target state, so that the observation model becomes:

$$g(z \mid x, q) = g^s(z^s \mid x^s) g^m(z^m \mid x^m, q). \tag{3}$$

An example of a metric part could be the range and bearing to a target, equivalent to the measurement models in standard non-semantic mapping and tracking tasks. The class part is represented by a confusion matrix which describes the probability of detecting class z^c conditioned on the true class x^s. This takes the form of a confusion matrix where each row matrix represents the instances of the true class while each column represents the instances of the measured class. For example, the entry in row 2 column 4 represent the probability of measuring class 4 given that the true target is of class 2. Mathematically, this is a right stochastic matrix. Finally, the **false positive** (or clutter) measurements are modeled by the clutter PHD, $\gamma(z \mid q)$, which describes both the number and locations (in measurement space) of the clutter measurements. As with the detection model, in this paper we assume that this is independent of the class, though nothing about the theory of the SPHD filter requires this to be the case.

These three target models and three sensor models are all necessary to utilize the (S)PHD filter In practice, the user can either specify the models based on experience/intuition or learn models in a data-driven manner, as we have done numerous times in the past [7–9]. We have found that obtaining accurate detection and clutter models is essential to obtaining a correct target estimate. If these models do not accurately reflect the true behavior of the sensor then often the PHD will contain the correct number of peaks but the weight in each peak will not be close to 1. As a result, in practice we have found that counting the number of peaks in the final PHD to be a more reliable estimate of the target number than integrating the PHD.

2.3 SPHD Prediction and Update Steps

Using these target and sensor models from above, the SPHD filter prediction and update equations are:

$$\bar{v}^t(x) = b(x) + \int_E f(x \mid \xi) p_s(\xi) v^{t-1}(\xi) \, d\xi \tag{4}$$

$$v^t(x) = \big(1 - p_d(x \mid q)\big) \bar{v}^t(x) + \sum_{z \in Z_t} \frac{\psi_{z,q}(x) \bar{v}^t(x)}{\eta_z(\bar{v}^t)} \tag{5}$$

$$\eta_z(v) = \gamma(z \mid q) + \int_E \psi_{z,q}(x) v(x) \, dx \tag{6}$$

$$\psi_{z,q}(x) = g(z \mid x, q) p_d(x \mid q), \tag{7}$$

where $\psi_{z,q}(x)$ is the probability of a sensor at q receiving measurement z from a target with state x. Note that these take the equivalent form to those of the standard PHD filter, except that in our case both the target state space and measurement space include a discrete class label from the set C. The SPHD filter recursively applies (4) and (5) to track the first order statistical moment of RFS for each target.

The addition of the discrete label space offers advantages beyond simply providing a mechanism to track the type of object. Due to the mathematical form of the PHD, the standard PHD filter does not perform well when targets are densely clustered. When a group of targets are close (compared to the sensor noise), all of the targets would appear as one combined peak in the density function rather than being separate discrete peaks. However, the SPHD filter provides a way to separate targets out based on the class label. For example, a person seated on a chair next to a desk in front of a computer could show up as 4 distinct targets with separate class labels in the SPHD filter instead of a single peak of size 4 in the standard PHD filter. Adding in the semantic information will help because it provides a way to separate out targets of different types.

While we do not consider it in this paper, one could consider the possibility of switching between classes using the transition model. For example, a person could switch from being seated to walking, if someone wants to consider those as two separate classes.

2.4 The Parallel PHD Filters Method

As a point of comparison, we will test the SPHD filter against the standard PHD filter. In the latter case, we will have multiple PHD filters running in parallel, one for each target class. Each of these separate PHD filters will use the target models for their respective classes. The measured class will be used to funnel the measurements to their respective PHD filters, which will use class-agnostic sensor models. In particular, the observation model g of the standard PHD filter only contains the metric portion g^m. This implies that the filters completely trust the observed class, which is a reasonable assumption when the confusion matrix is close to the identity. Most of the semantic mapping work makes the same assumption. We will later on compare the parallel PHD filters method with our proposed SPHD filter method.

3 Simulations

We demonstrate the results by a series of simulations using ROS Kinetic running on Ubuntu 16.04. All these simulations are using only one robot for simplicity, though multiple robots can also work together for target search [10]. For simplicity, in each simulation we use only one robot to search for a small number of (static and/or dynamic) targets. There is no impediment to using the SPHD with multiple robots, which we have previously shown with the standard PHD filter [10]. The number of target types is similarly not constrained in theory so long as the robot is equipped with a classification algorithm to detect each class. For example, instead of simply using the class "person," as we do in our experiments, a robot could distinguish between people in different states, *e.g.,* "person sitting," "person standing," "person walking," and others.

The robot model we use is a differential drive robot with a maximum linear velocity of 0.4 m/s and a maximum angular velocity of 1.2 rad/s. The PHD is represented by a uniform grid of particles [29] with a resolution of 0.2 m. The initial weight of each particle is identical, meaning that the targets are uniformly likely to be appear in the environment.

3.1 OSPA Error

We measure the error between the estimated target set and the true target set using the Optimal SubPattern Assignment (OSPA) metric [26]. The error between two sets X, Y, where $|X| = m \leq |Y| = n$ without loss of generality, is

$$d(X,Y) = \left(\frac{1}{n} \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_\alpha(x_i, y_{\pi(i)})^p + \alpha^p (n-m) \right)^{1/p}, \tag{8}$$

where c is a cutoff distance, $d_\alpha(x,y) = \min(\alpha, \|x - y\|)$, and Π_n is the set of all permutations of the set $\{1, 2, \ldots, n\}$. OSPA finds the lowest cost assignment, where elements $x \in X$ and $y \in Y$ can be matched only if they are within distance α of each other. This can be efficiently computed using the Hungarian algorithm [14, 21]. We use $\alpha = 10$ m and $p = 1$.

The OSPA error describes the average error in the target positions with a maximum per target error of α (which is 10 in this work). Given that, when a target is found there is typically a drop in the OSPA of around $10/n$ (if there are n targets), indicating that the error for that target went from 10 to around 0. Targets are precisely tracked when the OSPA error is getting closed to zero.

To extract an estimated target set, we take advantage of the grid structure of the PHD. We use a convolution operation to identify the local maxima over a 5×5 grid of particles (1×1 m area) for each type. We then discard any local maxima that do not have a sufficiently high weight.

3.2 Stationary targets

Our first scenario will test the SPHD filter's ability to track multiple types of stationary targets. We conduct stationary target simulations in a 40×30 m map with 6 rooms and one corridor, as Fig. 1 shows. The robot follows a predefined route through the environment, traversing this route twice during each trial (for a total approximate travel time of 1900 s). This path ensures that the robot see the entire environment twice to observe the differences between first seeing an object and re-observing it later.

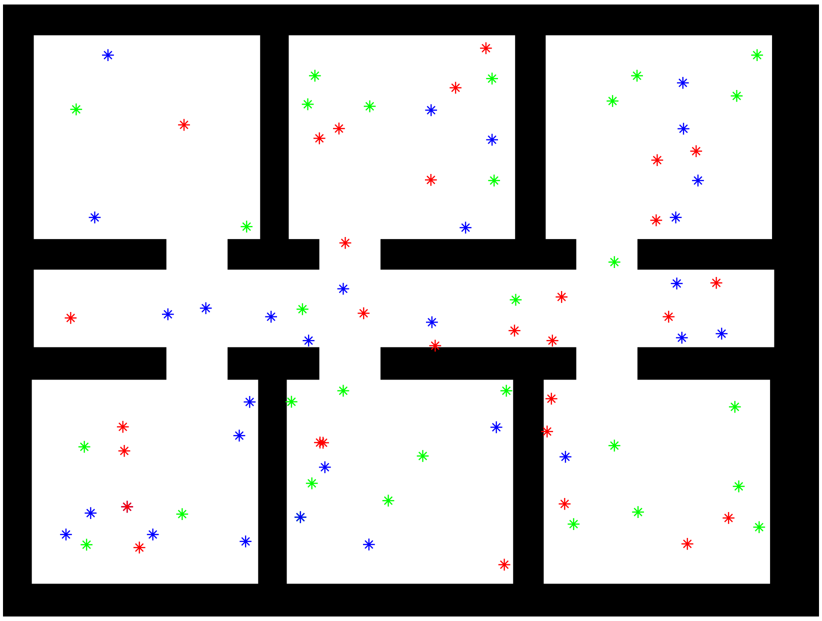


Fig. 1. The 40 × 30 m environment for stationary target search. Markers show an example of target distribution of 3 classes: person (red), chair (green) and table (blue).

Target Models There are three classes of targets: person, chair, and table, with 30 targets of each type. Targets are randomly distributed in the environment. Chairs and tables are static targets. When searching for static targets, the target motion model is the identity map, the survival probability is unity, and the birth PHD is zero. This was true for both the ground truth motion of the targets and the models used by the robots in the PHD prediction equation (4).

In general, people may enter or leave the environment and may move about within the environment. However, in this first test all people remain stationary so the ground truth model is the same as for chairs and tables. This does not match the motion model used in the PHD filter, which was a truncated Gaussian random walk with spherical covariance matrix with standard deviation 0.01 m per time step (0.1 s). The probability of survival and the birth PHD were

$$p_s(x) = \begin{cases} \|x - \partial E\| & \|x - \partial E\| \leq 1\,\mathrm{m} \\ 1 & \text{else} \end{cases} \tag{9}$$

$$b(x) = 1.0 \cdot 10^{-4} \tag{10}$$

where ∂E is the boundary of the environment. When analyzing the results we will examine the effects of the mismatch in the true target models and those used in the SPHD filter.

Observed True	Person	Chair	Table
Person	0.9	0.05	0.05
Chair	0.1	0.8	0.1
Table	0.15	0.15	0.7

(a) Confusion matrix 1

Observed True	Person	Chair	Table
Person	0.8	0.15	0.05
Chair	0.2	0.7	0.1
Table	0.2	0.25	0.55

(b) Confusion matrix 2

Observed True	Person	Chair	Table
Person	0.7	0.15	0.15
Chair	0.15	0.6	0.25
Table	0.25	0.25	0.5

(c) Confusion matrix 3

Observed True	Person	Chair
Person	0.9	0.1
Chair	0.1	0.9

(d) Confusion matrix 4

Table 1. Confusion matrices for different trials.

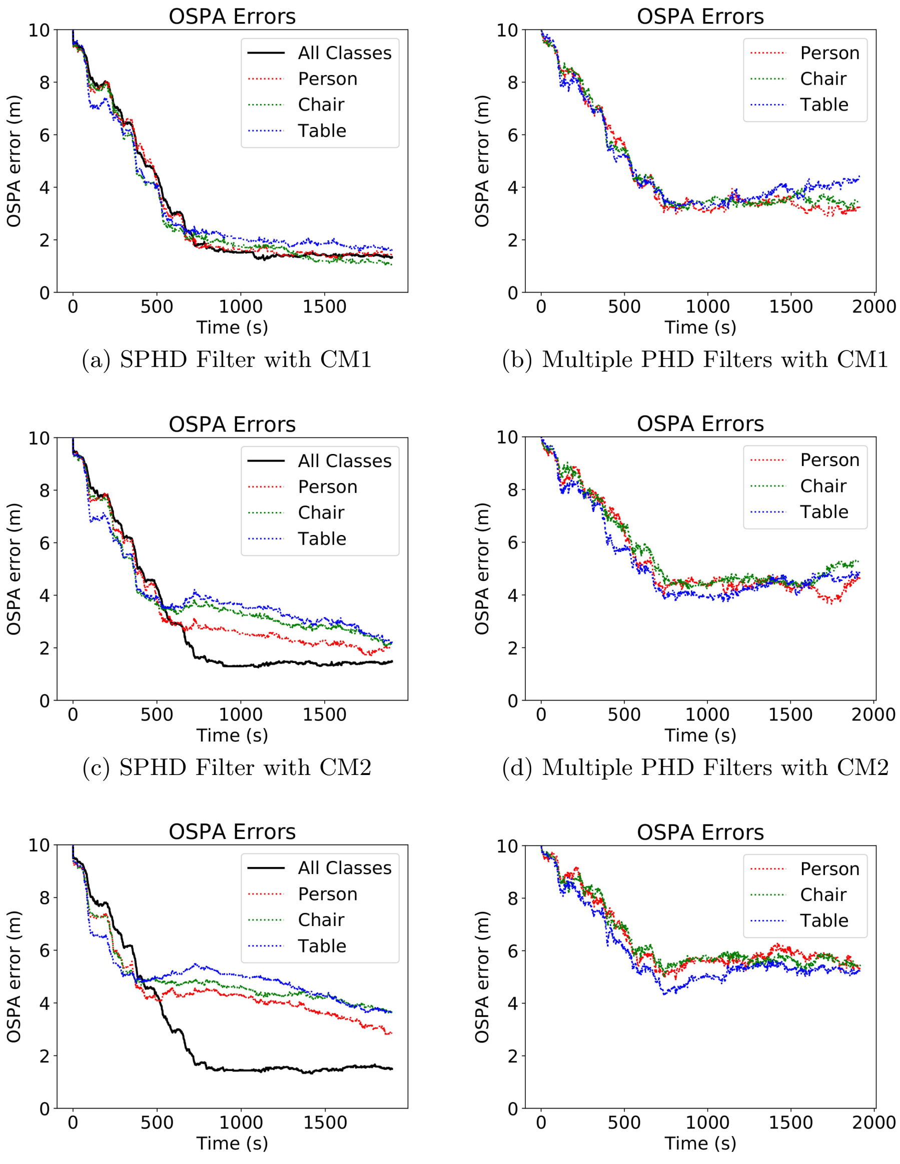
Sensor Models We assume that the robot carries an RGB-D camera with a forward-facing 120.0° field of view (FoV) and 5 m maximum detection range. This sensor returns the range and bearing to each detected target (the metric part z^m) and a class label for each target (the semantic part, z^s). The detection model and clutter of the sensor is shown as

$$p_d(x \mid q) = \begin{cases} 1 - 0.02\|x - q\| & x \text{ in FoV} \\ 0 & \text{else} \end{cases} \tag{11}$$

$$\gamma(z \mid q) = 1.5 \cdot 10^{-3} \tag{12}$$

The total expected number of clutter detections per measurement set, found by integrating the clutter PHD over the sensor FoV, is $\int \gamma(z \mid q)\, dz = 0.04$. We assume that the range-bearing (metric) measurement model $g^m(z^m \mid x^m, q)$ follows a multivariate Gaussian distribution with mean $\mu(x, q)$ (the position of the target in the robot's sensor frame) and diagonal covariance Σ (so that the noise of range and bearing measurements are independent). The standard deviation of the range and bearing noise are 0.02 m and 2.0 degrees respectively. We assume that the confusion matrix, $g^s(z^s \mid x^s)$, of the sensor detecting these three classes in this environment is described in Table 1a.

Results As we previously mentioned, we compare the SPHD filter, which simultaneously tracks all classes, to a set of parallel PHD filters method, which each track a single target type. For each method we use three different confusion matrices (CM), shown in Tables 1a–1c. We conduct 5 trials for each configuration (SPHD vs. parallel PHD and each confusion matrix). Each trial has a different target distributions. However, the target distributions are the same across different configurations, so, for example, trial 1 using the SPHD filter with CM1 has the same target configuration as trial 1 using parallel PHD filters with CM3. Figure 2 show the average OSPA errors for each class over all 5 trials.



(a) SPHD Filter with CM1

(b) Multiple PHD Filters with CM1

(c) SPHD Filter with CM2

(d) Multiple PHD Filters with CM2

(e) SPHD Filter with CM3

(f) Multiple PHD Filters with CM3

Fig. 2. Results of the two methods for stationary target tracking with different confusion matrices (CM) to classify targets. Each figure shows the average OSPA errors over 5 runs of tracking each of the three classes: person, chair and table. We also plot the class-agnostic OSPA error in the case of the SPHD filter. This is not available for the case of multiple PHD filters since there is no single PHD filter for all targets.

Figure 2a shows the results of using the SPHD filter with confusion matrix 1 (Table 1a), which has the highest classification accuracy. We see that the OSPA errors decrease in the first half of time since the robot keeps exploring new area in the environment. In the second half of time the robot passes through the environment for the second time, during which time the OSPA error fluctuates slightly due to the appearance of new clutter/missed detections, the correction of previous clutter/missed detections, and classification errors. We can see that all of the classes have a similar OSPA error throughout and that each of these is approximately the same as the class-agnostic OSPA error (All Classes label), which uses only metric information. This indicates that the tracking performance is not limited by classification error, but rather by other phenomena, such as clutter/missed detections or sensor noise. This is despite the fact that each class has only a 70–90% chance of being correctly identified on a per-frame basis.

Figure 2b shows the results of the same scenario using the parallel PHD filters. We can see that the OSPA error follows a similar trend, decreasing steadily during the first pass and then leveling out after that. However, the final OSPA error is significantly higher than with the SPHD. From these results we can see that the parallel PHD filters have a much more difficult time dealing with misclassifications.

Figures 2c–2f show the results with the other confusion matrices (Tables 1b and Table 1c), which have significantly higher rates of misclassification. We can see that the class-agnostic OSPA is very similar between all three SPHD tests. This indicates that all of the differences between the class-dependent OSPA lines are likely due to the differences in the confusion matrix. We can see that the SPHD filter follows a similar trend during the first 500 s in every case. After this, we see that the confusion matrix with a higher chance of misclassification has a higher OSPA, a very intuitive result. Despite this, the OSPA continues to steadily, if slowly, decrease (eventually reaching the class-agnostic levels in Figure 2c). This indicates that the SPHD filter is able to perform well even with high error rates, provided that it receives sufficient data. On the other hand, the parallel PHD filters do not show this trend. Instead, the OSPA error simply levels out and does not increase or decrease by a significant amount after about 750 s. Finally, the OSPA error in the case of the parallel PHD filters fluctuates more wildly than in the case of the SPHD filter. This is likely due to the SPHD filter's superior ability to handle uncertainty in the class of targets.

3.3 Moving targets

We also want to test the SPHD filter's ability to track a combination of static and mobile targets. In this case, the robot monitors an open 20×20 m environment. There are originally 10 people and 10 chairs. Just like the last test, chairs are stationary in both their ground truth motion and in their SPHD filter motion model. The robot uses the same motion model for people in the SPHD filter as in the static target case. However, instead of standing still, people are continuously moving at 0.3 m/s towards random waypoints, uniformly sampled from a 22×22 m area. This leaves 1 m outside of the environment for each boundary so that

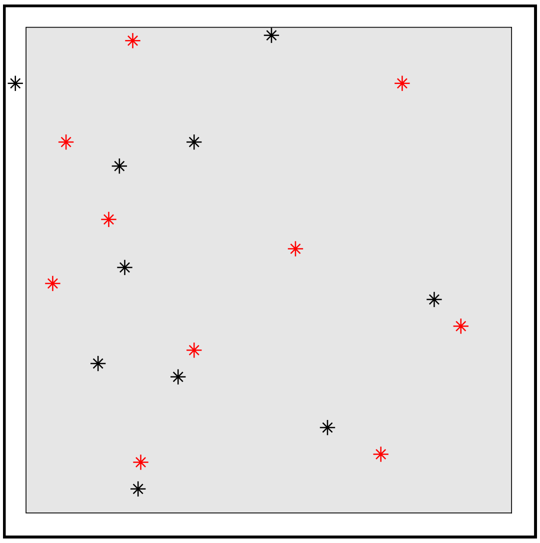


Fig. 3. Example of the 20×20 m environment (gray square) with markers showing the distribution of 2 classes: people (black) and chairs (red). Note that people may move anywhere within the 22×22 m area (white square), allowing them to enter and leave the observed environment.

people may occasionally leave or enter the robot's area. When a person reaches their destination, they select a new waypoint and repeat this process. Note that again there is a discrepancy between the true and modeled motion of the people. In particular, the true velocity of the people is 3 standard deviations from the mean, making this a very challenging tracking task.

The robot is placed statically in the middle of the environment with a sensor FoV that covers the whole environment. We make this choice because the focus of this work is to demonstrate the capabilities of the SPHD filter, not to develop a control strategy for target search and tracking. This will be left as future work, perhaps using some of the authors' previous work on target tracking controllers [7, 10]. The detection probability is 0.99 in the entire environment. The clutter model, γ, and the metric observation model, g^m, are identical to those from the static target case. Table 1d shows the confusion matrix assumption of the sensor classifying these four classes in this environment.

Figure 4a shows the resulting OSPA error from our trials. During the 350 s of searching, both static and dynamic targets are well tracked most of time. Compared with the stationary target tracking test using CM1 (Table 1a), where the probability of a person being classified correctly is also 0.9, the OSPA error fluctuates more and the overall OSPA error is a little higher. This is not surprising given that the targets are now moving and there is a larger difference between the true and assumed motion model for the people.

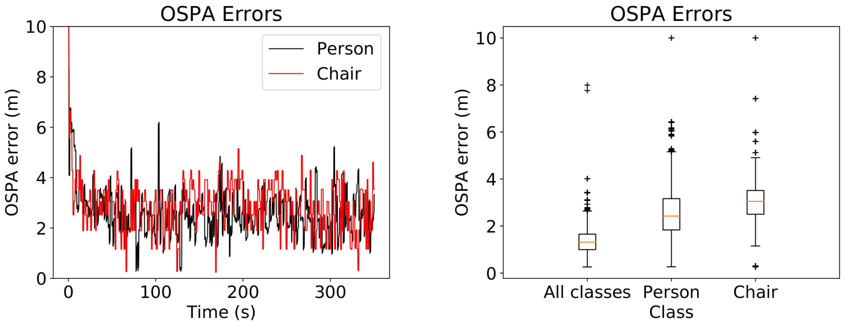


Fig. 4. Result of the SPHD filter tracking both dynamic and static targets. Figure 4a shows the OSPA errors of person and chair class over 350 s. Figure 4b shows boxplots of OSPA errors of both classes as well as all targets.

Figure 4b shows boxplots of the OSPA errors, providing a clearer view of the steady-state behavior. We can see that the chair class has a lower standard deviation compared to the person class, which is not surprising given that the true and assumed motion models for the chair match. Also, the median OSPA error is lower for the person that it is for the chair class, though not significantly. Both of these medians are higher (by about 1) than the class-agnostic OSPA. Given the definition of the OSPA error and the fact that there are 10 targets of each type, this means that the SPHD filter is misclassifying one target of each type.

3.4 Computation time

We conducted our simulations on a workstation equipped with a 3.7 GHz Intel Xeon E3-1240 v6 and 16 GB of RAM and we implemented the SPHD filter in C++ using ROS libraries. In our trials, each recursion of both the PHD filter and the SPHD filter took approximately 5–10 ms per class, depending on the number of particles within the sensor field of view, the number of measurements received, and also on the other processes concurrently running on the computer. We did not see any significant difference between the time per class using the parallel PHD filters versus the single SPHD filter.

Extrapolating from these results, we could expect real-time operation in these scenarios using sensors that receive data at 30–50 Hz. This is at or above the frame rate of most image-based sensors. However, in practice, most processors available onboard mobile robots are less capable than the Xeon that we used. To address this, there are two easy avenues to improve the efficiency of our code which we will explore as we work towards validating the SPHD filter in hardware. First, we could use the `CMAKE_BUILD_TYPE=Release` option when compiling our code to create a more optimized executable compared to the standard `Debug` option that we used when developing and testing our code. Second, the PHD update step is highly parallelizable, so one could use multi-threading or GPU-based computation to significantly decrease run time. Finally, deploying our system in hardware will likely require the use of an image-based classification algorithm. These tend to run more slowly than the SPHD filter updates, and thus we do not expect the SPHD filter to be the computational bottleneck in the perception and estimation pipeline.

4 Conclusion

In this paper we propose the semantic PHD filter algorithm, a RFS-based multi-target tracking algorithm which uses both metric and semantic information to simultaneously track multiple classes of targets. Mathematically, the key to defining the SPHD filter is to augment both the target state space and the measurement space with a discrete set of class labels. The various target and sensor models within the standard PHD filter framework utilize this additional label state to differentiate between target types. Some models, like the target motion model, are defined separately for each individual class while others, like the observation model, must contain a single model for all target types. Using these models, the SPHD filter can then iteratively propagate the PHD and handle uncertainty, such as the possibility of target misclassification, in a theoretically principled manner.

We conduct a series of simulations using multiple target types, including a mixture of static and dynamic targets, to demonstrate the performance of the SPHD filter. In all cases, the SPHD filter outperforms a system that utilizes multiple standard PHD filters (one for each class) in parallel. In particular, the SPHD filter demonstrates an ability to recover from prior misclassifications, even when the probability of correct classification is barely over 50%. Given this, we expect the SPHD filter to perform well in real-world experiments, which is one direction of future work. Other directions include using multiple robots to search for and track multi-class targets in a coordinated manner and applying an active control algorithm to enable one or more robots to search for and track dynamic targets.

5 Acknowledgements

This work was funded by NSF grant IIS-1830419 and the Amazon Research Awards program. We would like to thank Zhijia Chen from Temple University for assistance with the data processing work.

References

1. Atanasov, N., Zhu, M., Daniilidis, K., Pappas, G.J.: Semantic localization via the matrix permanent. In: Robotics: Science and Systems, vol. 2 (2014)

2. Bahlmann, C., Zhu, Y., Comaniciu, D., Köhler, T., Pellkofer, M.: Method for combining boosted classifiers for efficient multi-class object detection (2010). US Patent 7,769,228
3. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: CVPR 2011, pp. 2025–2032. IEEE (2011)
4. Blackman, S.S.: Multiple hypothesis tracking for multiple target tracking. IEEE Aerosp. Electron. Syst. Mag **19**(1), 5–18 (2004)
5. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1722–1729. IEEE (2017)
6. Daley, D.J., Vere-Jones, D.: An introduction to the theory of point processes, vol. 1. Springer (2003)
7. Dames, P., Kumar, V.: Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. IEEE Transactions on Automation Science and Engineering **12**(3), 850–864 (2015)
8. Dames, P., Kumar, V.: Experimental characterization of a bearing-only sensor for use with the PHD filter. arXiv preprint arXiv:1502.04661 (2015)
9. Dames, P., Tokekar, P., Kumar, V.: Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. The International Journal of Robotics Research **36**(13-14), 1540–1553 (2017). DOI 10.1177/0278364917709507
10. Dames, P.M.: Distributed multi-target search and tracking using the PHD filter. Autonomous Robots (2019). DOI 10.1007/s10514-019-09840-9
11. Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE J. Oceanic Eng. **8**(3), 173–184 (1983)
12. Gálvez-López, D., Salas, M., Tardós, J.D., Montiel, J.: Real-time monocular object SLAM. Robotics and Autonomous Systems **75**, 435–449 (2016)
13. Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: A survey. Robotics and Autonomous Systems **66**, 86–103 (2015)
14. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
15. Lee, C.S., Clark, D.E., Salvi, J.: SLAM with dynamic targets via single-cluster PHD filtering. IEEE Journal of Selected Topics in Signal Processing **7**(3), 543–552 (2013)
16. Lin, L., Bar-Shalom, Y., Kirubarajan, T.: Track labeling and PHD filter for multitarget tracking. IEEE Transactions on Aerospace and Electronic Systems **42**(3), 778–795 (2006)
17. Mahler, R.: Multitarget Bayes filtering via first-order multitarget moments. IEEE Trans. Aerosp. Electron. Syst. **39**(4), 1152–1178 (2003)
18. Mahler, R.: Statistical multisource-multitarget information fusion, vol. 685. Artech House Boston (2007)
19. Moratuwage, D., Wang, D., Rao, A., Senarathne, N., Wang, H.: RFS collaborative multivehicle SLAM: SLAM in dynamic high-clutter environments. IEEE Robotics & Automation Magazine **21**(2), 53–59 (2014)
20. Mullane, J., Vo, B.N., Adams, M.D., Vo, B.T.: A random-finite-set approach to bayesian SLAM. IEEE Transactions on Robotics **27**(2), 268–282 (2011)
21. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957)
22. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. Robotics and Autonomous Systems **56**(11), 915–926 (2008)

23. Pronobis, A., Jensfelt, P.: Large-scale semantic mapping and reasoning with heterogeneous modalities. In: 2012 IEEE International Conference on Robotics and Automation, pp. 3515–3522. IEEE (2012)
24. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
25. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: SLAM++: Simultaneous localisation and mapping at the level of objects. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1352–1359 (2013)
26. Schuhmacher, D., Vo, B.T., Vo, B.N.: A consistent metric for performance evaluation of multi-object filters. IEEE Trans. Signal Processing **56**(8), 3447–3457 (2008)
27. Stone, L.D., Streit, R.L., Corwin, T.L., Bell, K.L.: Bayesian multiple target tracking. Artech House (2013)
28. Thrun, S., Leonard, J.J.: Simultaneous localization and mapping. Springer handbook of robotics pp. 871–889 (2008)
29. Vo, B.N., Singh, S., Doucet, A., et al.: Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. In: Proc. Intl Conf. on Information Fusion, pp. 792–799 (2003)
30. Vo, B.N., Vo, B.T., Phung, D.: Labeled random finite sets and the Bayes multi-target tracking filter. IEEE Transactions on Signal Processing **62**(24), 6554–6567 (2014)
31. Vo, B.T., Vo, B.N.: Labeled random finite sets and multi-object conjugate priors. IEEE Transactions on Signal Processing **61**(13), 3460–3475 (2013)
32. Zender, H., Mozos, O.M., Jensfelt, P., Kruijff, G.J., Burgard, W.: Conceptual spatial representations for indoor mobile robots. Robotics and Autonomous Systems **56**(6), 493–502 (2008)