Algorithm for Searching and Tracking an Unknown and Varying Number of Mobile Targets using a Limited FoV Sensor

Yoonchang Sung and Pratap Tokekar

Abstract— **We study the problem of searching and tracking a collection of moving targets using a robot with a limited Field-of-View (FoV) sensor. The actual number of targets present in the environment is not known a priori. We propose a search and tracking framework based on the concept of Bayesian Random Finite Sets (RFSs). Specifically, we generalize the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter which was previously applied for only tracking problems to allow for simultaneous search and tracking. The proposed framework can extract individual target tracks as well as estimate the number and spatial density of the targets. We also show how to use Gaussian Process (GP) regression to extract and predict non-linear target trajectories in this framework. We demonstrate the efficacy of our techniques through representative simulations where we also compare the performance of two active control strategies.**

I. INTRODUCTION

We study the problem of searching for and tracking a set of targets using a robot with a limited FoV sensor. This problem is motivated by robotic search-and-rescue [1], [2], surveillance [3], crowd/traffic monitoring [4], [5], and wildlife habitat monitoring [6]–[8]. We specifically consider scenarios where the number of targets being searched is not known a priori. The targets may move during the search process and the motion model of the targets is not known exactly. As the targets are mobile, the robot is also tasked with tracking the target trajectories.

The search and tracking problems can be loosely distinguished depending on whether or not a target is in the FoV: *tracking* when targets are in the FoV, and *search* when targets are out of the FoV. Once all targets are observed by sensor platforms, the search task is accomplished. To successfully conduct the tracking task, the states of targets must be estimated at each time and trajectories of individual targets must be maintained over time. A robust tracking technique must be able to deal with clutter (false positive) measurements which is especially challenging since the true number of targets is not known.

Search techniques have been applied to a broad range of problems (*e.g.*, [9]–[14]). The recent survey by Chung et al. [15] gives a good overview of the search problem.

For the multitarget tracking problem, Joint Probability Data Association (JPDA) [16] and Multiple Hypothesis Tracking [17] have become canonical algorithms. These techniques have been applied to many problems including human following [18], object tracking [19] and human-robot interaction [20]. However, JPDA requires solving the data association problem which is especially costly when the actual number of targets is not known exactly [21]. Conventional Bayes trackers use a vector representation in which the order of the targets and its size is known and fixed. This makes tracking with an unknown number of targets intractable. However, the Probability Hypothesis Density (PHD) filter [22] that we use in this paper avoids these problems with the help of random set representations [23].

Several techniques have been proposed to unify the search and tracking problems [1], [24]. These include the sequential Monte Carlo filter [25], [26] as well as the PHD filter [4]. However, the existing works focus on estimating the number of targets and their spatial densities but cannot estimate trajectories of individual targets. On the other hand, there are existing works on estimating individual target trajectories but assuming unlimited FoV [21]. Our main contribution is to generalize the tracking algorithms for unlimited FoV sensing to the case of limited FoV. We also show how to extend the tracking to non-linear motion models by leveraging GP regression [27] based on the prior work in [4], [28].

The rest of the paper is organized as follows. We begin by describing the problem setup in Section II. We present a brief introduction to GM-PHD in Section III. Our proposed algorithm is presented in Section IV. We present results from representative simulations in Section V before concluding with a discussion of future work in Section VI.

II. PROBLEM DESCRIPTION

Our input is an estimate of the number of targets and a probability distribution over their initial spatial locations. However, the actual number of targets may be different. We assume that all targets move independently of each other. The targets may move on non-linear trajectories. However, we assume that the trajectories are smooth (in the sense, that will become clearer in Section IV-B). If a target is present in the FoV, then it is detected by the robot with probability p_D. If the target is detected, then the sensor returns a noisy measurement of the position of the target. We assume that the measurement noise is additive and Gaussian. In addition, at any time step, the sensor may also generate clutter (false positive) measurements following a Poisson RFS [29].

The proposed search and tracking framework is based on the concept of RFSs. The proposed method can estimate the states of targets and the number of targets at the same time, initiate and terminate individual target tracks, and distinguish between the search and tracking functions. In this paper,

Y. Sung and P. Tokekar are with the Department of Electrical and Computer Engineering, Virginia Tech, USA. {yooncs8, tokekar}@vt.edu.

This material is based upon work supported by the National Science Foundation under Grant No. 1637915.

we present illustrations and simulations assuming that the environment is 2D and obstacle-free, and the robot has a circular FoV. However, the proposed techniques easily extend to more complex scenarios.

III. BACKGROUND ON BAYESIAN RFS

Recursive Bayesian Estimation (RBE) is a canonical tool to estimate target states from noisy sensor observations. A standard assumption is that the number of targets is known exactly. Hence, we can treat the positions of all the targets at any time as a random vector and use RBE for estimation. For the setting considered in this paper, standard RBE techniques cannot directly be used since there is uncertainty on the length of the random vector itself. Mahler [22] developed the PHD filter to tractably solve exactly this class of problems. The PHD, also known as the intensity function, is the first-order statistical moment of a RFS. When integrated over any subset of the environment, it yields the expected number of targets present in that subset. The advantage of PHD is that it estimates both target states and the number of targets simultaneously without the necessity of data association. We briefly discuss the PHD filter next and refer the reader to reference [29] for more details.

We denote the PHD by v and the multitarget posterior density by $p_{k|k}(X|Z_k)$. X and Z are a multitarget state set and observation set, respectively. The expected number of targets within any region S is equal to the integral of intensity of target state vectors over S. That is,

$$\int |X \cap S| p_{k|k}(X|Z_k) \delta X = \int_S v(\mathbf{x}) d\mathbf{x}, \tag{1}$$

We define $v_{k|k-1}(\mathbf{x}) := v_{k|k-1}(\mathbf{x}|Z_{k-1})$ and $v_{k|k}(\mathbf{x}) := v_{k|k}(\mathbf{x}|Z_k)$ for notational ease. The prediction step of the PHD recursion is given by,

$$\begin{aligned} v_{k|k-1}(\mathbf{x}) = &\int p_S(\mathbf{w}) f_{k|k-1}(\mathbf{x}|\mathbf{w}) v_{k-1|k-1}(\mathbf{w}) d\mathbf{w} + \\ &\int \omega_{k|k-1}(\mathbf{x}|\mathbf{w}) v_{k-1|k-1}(\mathbf{w}) d\mathbf{w} + \beta_k(\mathbf{x}). \end{aligned} \tag{2}$$

$p_S(\cdot)$, $f_{k|k-1}(\cdot|\cdot)$, $\omega_{k|k-1}(\cdot|\cdot)$ and $\beta_k(\cdot)$ denote the probability of survival of existing targets, the transition model, the intensity of spawning new targets from existing ones, and the intensity of birth targets. The update step is given by,

$$\begin{aligned} v_{k|k}(\mathbf{x}) = &[1 - p_D(\mathbf{x})] v_{k|k-1}(\mathbf{x}) + \\ &\sum_{\mathbf{z} \in Z_k} \frac{p_D(\mathbf{x}) g_k(\mathbf{z}|\mathbf{x}) v_{k|k-1}(\mathbf{x})}{k(\mathbf{z}) + \int p_D(\mathbf{w}) g_k(\mathbf{z}|\mathbf{w}) v_{k|k-1}(\mathbf{w}) d\mathbf{w}}. \end{aligned} \tag{3}$$

$p_D(\cdot)$, $g_k(\cdot|\cdot)$ and $k(\cdot)$ denote the probability of detection, the sensor likelihood model, and the intensity of clutters.

The PHD filter propagates the posterior intensity recursively over time, as in Equations 2 and 3. The exact derivation follows from reference [22].

Two approaches have gained substantial attention for the realization of PHD: the particle PHD [30] and the GM-PHD [21] filters. Particle PHD is suitable for dealing with non-linear motion of targets, whereas GM-PHD assumes that a target has a linear motion model. Nevertheless, we can use the Extended Kalman Filter or Unscented Kalman Filter versions of GM-PHD as presented in [21] for non-linear motion models. GM-PHD gives a closed-form solution without needing large sample size or clustering techniques to extract multitarget state estimates, unlike particle PHD. In a GM-PHD the intensity function is represented as a Gaussian mixture model of one or more Gaussian components (Figure 1). Each Gaussian component is represented by its mean, covariance, and weight. The weight of a component gives the expected number of targets generated as a result of that component. We refer the reader to reference [21] for a detailed discussion of GM-PHD.

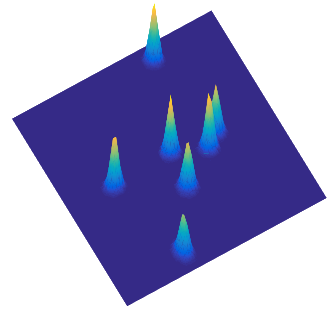


Fig. 1. The GM-PHD filter with 7 Gaussian components

IV. PROPOSED GM-PHD SEARCH AND TRACKING ALGORITHM

Figure 2 gives an overview of the proposed algorithm. We start with an initial estimate of PHD. Multitarget Bayes filter, *i.e.*, the prediction and update steps, is applied recursively to estimate both search and tracking targets. Since PHD is employed, additional data association between targets and measurements is not required. The pruning and merging schemes reduce the number of Gaussian components with low and similar weights, respectively. Then, multitarget state estimates are extracted from GM-PHD and used for maintaining trajectory states of targets. Estimates of the targets in the search region are allowed to enter the tracking regions and vice versa using boundary condition. Finally, an active control strategy is used to get the trajectory for the robot.

We describe our new contributions in this section and refer the reader to reference [31] for a description of the other blocks in Figure 2. Specifically, we show how to extend GM-PHD to allow for separate multitarget search and tracking states (Section IV-A), how to use GP regression to predict non-linear motion models of the targets (Section IV-B), how to extract and manage tracks of individual targets (Section IV-C), how to handle targets moving from search to tracking states (Section IV-D), and present two naive strategies for actively controlling the robot's state (Section IV-E).

A. Multitarget State and Observation Spaces

Tracking and Search States: The true but unknown state of a target is represented as $\mathbf{x}_i = (x_1, ..., x_d, i) \in \chi \times \mathbb{Z}_{\geq 0}$, where $\chi \subseteq \mathbb{R}^d$ is a d-dimensional environment and $i \in \mathbb{Z}_{\geq 0}$ is a non-negative integer denoting the target ID. A multitarget

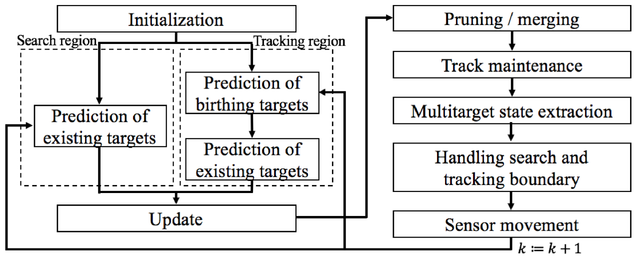


Fig. 2. Flowchart for GM-PHD search and tracking algorithm.

state set X is defined as $X = \{\mathbf{x}_i \mid \forall i \in \{1, ..., n\}\}$. We can partition X into two disjoint subsets,

$$X = X^T \cup X^S, \tag{4}$$

where X^T and X^S denote the multitarget tracking and search states, respectively. X^T and X^S depend on the state of the robot at a given time k. X_k^T represents the set of states of targets that can be reliably detected in the FoV of the robot at time k. X_k^S represents the set of states of all other targets.

Let $\mathbf{y}_k$ denote the state of the sensor at time k. We assume that a target with true state $\mathbf{x}_{k,i}$ is detected by the robot with probability $p_D(\mathbf{x}_{k,i}, \mathbf{y}_k)$. We have,

$$\begin{aligned} X_k^T = &\{\mathbf{x}_{k,i} \mid \forall i \in \{1, ..., n_k\} \\ &\text{and } p_D(\mathbf{x}_{k,i}, \mathbf{y}_k) \geq \gamma\} \subset \chi, \end{aligned} \tag{5}$$

where γ is a desired threshold on the probability of detection. γ can depend on the FoV of the sensor. n_k is the true but unknown number of targets at time k. Likewise, a multitarget search state is given by,

$$\begin{aligned} X_k^S = &\{\mathbf{x}_{k,i} \mid \forall i \in \{1, ..., n_k\} \\ &\text{and } p_D(\mathbf{x}_{k,i}, \mathbf{y}_k) < \gamma\} \subset \chi. \end{aligned} \tag{6}$$

We denote the multitarget tracking and search state regions by χ^T and χ^S, respectively.

Measurement Model: At every time step, the robot gets a noisy observation of the true multitarget tracking state X_k^T. Let Σ_k be this random observation set which is a finite subset of the observation space $\mathcal{Z}$. $|\Sigma_k|$ is not necessarily equal to $|X_k^T|$ since some of the targets may not be detected (false negative), whereas some clutter measurements may be observed (false positive). Formally,

$$\Sigma_k = D(X_k) \cup C, \tag{7}$$

where $D(X_k) = \bigcup\limits_{i=1}^{n_k} D(\mathbf{x}_{k,i})$ selects measurements based on the probability of detection at time k and C denotes RFS of clutter. D models if a target in X_k^T is detected by the sensor as well as the additive Gaussian noise in the state measurement. C is the clutter drawn from a Poission RFS restricted to only tracking regions.

Bayesian RFS Estimation: Our goal is to predict and estimate the true multitarget tracking and search states, X_k^T and X_k^S, at every time step using the noisy observation sets. As described previously, we use GM-PHD Bayesian RFS techniques proposed in [21] for the estimation and extend it to allow updates of the tracking and search states individually. Analogous to the Kalman filter, the Bayesian RFS recursion predicts a RFS $\Xi_{k|k-1}$ and corrects the RFS estimate $\Xi_{k|k}$ at every time step.

The predicted RFS can be partitioned into two subsets,

$$\Xi_{k|k-1} = \Xi_{k|k-1}^T \cup \Xi_{k|k-1}^S, \tag{8}$$

where $\Xi_{k|k-1}^T$ and $\Xi_{k|k-1}^S$ are the predicted RFSs of multitarget tracking and search states, respectively. The predicted RFS of a multitarget tracking state has three terms:

$$\Xi_{k|k-1}^T = S^T(X_{k-1}^T) \cup W(X_{k-1}^T) \cup B, \tag{9}$$

where $S^T(X_{k-1}^T) = \bigcup_{i=1}^{n_{k-1}^T} S^T(\mathbf{x}_{k-1,i})$ models the motion model and survival of existing tracking targets, and $W(X_{k-1}^T) = \bigcup_{i=1}^{n_{k-1}^T} W(\mathbf{x}_{k-1,i})$ models spawning new targets from existing tracking targets, both given a multitarget tracking state at time $k-1$ with the number of tracking targets n_{k-1}^T. B denotes RFS of the target birth.

Since search targets are not observable, the spawning and birth models can be excluded from the predicted RFS:

$$\Xi_{k|k-1}^S = S^S(X_{k-1}^S), \tag{10}$$

where $S^S(X_{k-1}^S) = \bigcup_{i=1}^{n_{k-1}^S} S^S(\mathbf{x}_{k-1,i})$ and n_{k-1}^S is the number of search targets.

We adapt the prediction of PHD (Equation 2) and the update of PHD (Equation 3) to include the concept of a search region. The predicted PHD of Equation 2 with respect to Equations 9 and 10 can be computed by,

$$\begin{aligned} &v_{k|k-1}(\mathbf{x}) = \int_{\chi^T} p_S(\mathbf{w}) f_{k|k-1}(\mathbf{x}|\mathbf{w}) v_{k-1|k-1}(\mathbf{w}) d\mathbf{w} + \\ &\int_{\chi^T} \omega_{k|k-1}(\mathbf{x}|\mathbf{w}) v_{k-1|k-1}(\mathbf{w}) d\mathbf{w} + \beta_k(\mathbf{x}|\mathbf{x} \in \chi^T) + \\ &\int_{\chi^S} f_{k|k-1}(\mathbf{x}|\mathbf{w}) v_{k-1|k-1}(\mathbf{w}) d\mathbf{w}. \end{aligned} \tag{11}$$

While tracking targets in χ^T has birth and spawning models, the PHD of search targets in χ^S is only propagated by the transition density function. Note that the survival model in χ^S does not have $p_S(\cdot)$ term. The updated PHD of Equation 3 with respect to Equation 7 is

$$\begin{aligned} &v_{k|k}(\mathbf{x}) = [1 - p_D(\mathbf{x}|\mathbf{x} \in \chi^T)] v_{k|k-1}(\mathbf{x}|\mathbf{x} \in \chi^T) + \\ &\sum_{\mathbf{z} \in Z_k} \frac{p_D(\mathbf{x}|\mathbf{x} \in \chi^T) g_k(\mathbf{z}|\mathbf{x} \in \chi^T) v_{k|k-1}(\mathbf{x}|\mathbf{x} \in \chi^T)}{k(\mathbf{z}) + \int_{\chi^T} p_D(\mathbf{w}) g_k(\mathbf{z}|\mathbf{w}) v_{k|k-1}(\mathbf{w}) d\mathbf{w}} + \\ &v_{k|k-1}(\mathbf{x}|\mathbf{x} \in \chi^S). \end{aligned} \tag{12}$$

Since no measurements can be made in χ^S, Equation 12 is the same as Equation 3 except for the last term. Even though the covariance of search targets increases due to no correction being applied, we intentionally do not change the weight of the search targets.

B. Prediction using GP Regression

The PHD prediction (Equation 11) requires knowing the motion model, $f_{k|k-1}$, for each of the targets. In previous works, a simple linear motion model was applied [21]. Instead, we use GP regression [27] which is a non-parametric, Bayesian, and non-linear regression technique which requires specifying a kernel function. In our previous works, we have shown how GP regression can be employed to learn the spatial velocity vectors of targets for a real-world taxi dataset [4]. Here, we employ GP regression to extrapolate each target's trajectory and predict its future positions.

The hyperparameters for the kernel are learned offline using a training set consisting of noisy observations of the target's motion. Noisy measurements of the state of the targets are fed as input to GP regression, which produces a prediction of its future positions. In particular, we use GP regression to estimate d functions, $f_i(t)$ where $i = 1, \ldots, d$, that predicts the evolution of the state of the target along each of its d dimensions, independently. Figure 3 shows an example of the 2D case and the result of GP regression applied to a trajectory sample. From a distribution obtained from GP regression, future trajectory mean position with covariance can be extrapolated.

In order to apply GP regression to predict the motion of each Gaussian in GM-PHD, we must have a *confirmed track* of individual targets. If a Gaussian is not assigned to a *confirmed track*, then we can use a simple linear motion model for the prediction. Once a track is confirmed (*i.e.*, we have sufficient history of an individual target trajectory) we employ GP regression to predict its motion. The next subsection describes how to keep track of *confirmed tracks*.

C. Track Management

It is important to keep the track continuity of the PHD filter so that the trajectories of individual targets can be observed and maintained. N_k tracks at time k are denoted by $T_k = \{\mathbf{t}_q \mid \forall q \in \{1, ..., N_k\}\}$. The q-th track at time k, $\mathbf{t}_{q,k}$, is represented as: $\mathbf{t}_{q,k} = (x_{1,1}, x_{1,2}, ..., x_{1,d}, c_1, \ldots, x_{l,1}, x_{l,2}, ..., x_{l,d}, c_l, q) \subseteq \mathbb{R}^{d \times l} \times \mathcal{Z}_{\geq 0} \times \{0,1\}^l$, where d is the dimension of the target state, l is the life length of track, q is a non-negative integer representing the track ID, and c_i is a binary indicator denoting whether a target is a search or tracking target. The details of the track continuity of the particle PHD filter and GM-PHD filter are explained in [32] and [31], respectively.

We define two types of tracks: *confirmed tracks* and *tentative tracks*. When a target not assigned to any other existing tracks is detected, it generates a new *tentative track* with a new index q. After surviving for certain time period, this track is converted into a *confirmed track*. In this way, false positive tracks can be filtered out.

A *confirmed track* q, $\mathbf{t}_{q,k}$, at time k can be generated from a *tentative track* q, $\mathbf{t}_{q,k-1} = \{\mathbf{x}_r, \mathbf{x}_{r+1}, ..., \mathbf{x}_{k-2}, \mathbf{x}_{k-1}\}$, created at some time r when the following conditions are met: (1) A target i at time k, $\mathbf{x}_{i,k}$, is spawned from a target r at time $k-1$, $\mathbf{x}_{r,k-1}$, of track q, $\mathbf{t}_{q,k-1}$, where r is any arbitrary target ID; (2) The weight of target i at time k, $w_{i,k}$, satisfies $w_{i,k} > w_{TH}$, where w_{TH} is a threshold that determines the existence of a target from peaks of PHD; and (3) The life length of track q at time $k-1$, $l_{q,k-1}$, satisfies $l_{q,k-1} > l_{TH}$, where l_{TH} is a threshold on the maximum number of time steps to stay as a *tentative track*.

It is important to note that two different w_{TH} must be applied for extracting multitarget states in χ^S and χ^T. Also, once any *confirmed tracks* fall in χ^S, they permanently remain as *confirmed tracks* until detected again. There is no *tentative track* defined in χ^S.

D. Construction of Search Boundary using GP Regression

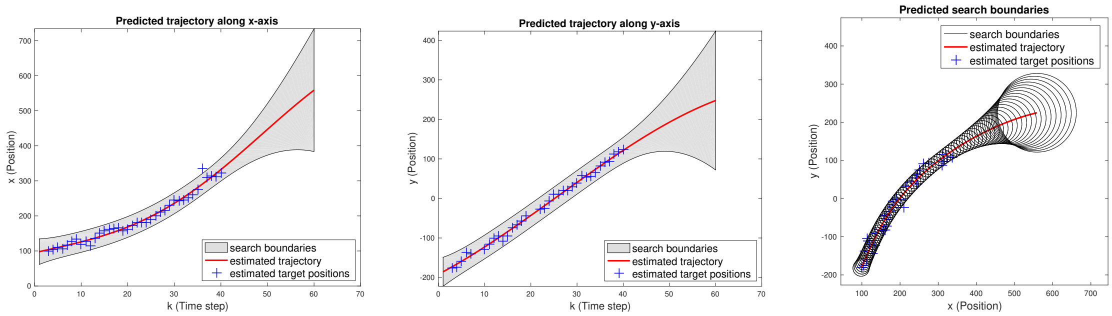
We refer to the region outside the FoV of the robot that is likely to contain targets as a *search boundary*. The terms search target and *search boundary* are used interchangeably because what is estimated for unobservable targets is an area that potentially contains a target of interest. Since targets in the search region are not observable, a naive initialization of a *search boundary* may lead to the failure of searching for targets. As soon as the sensor detects a new target, the corresponding weight of the target is increased to initiate a new tracking target in the tracking region. On the other hand, more careful means must be taken into account for a tracking target falling into the search region.

Figure 4 shows the process of constructing a *search boundary*. Every target that leaves the FoV creates a new *search boundary*. When a *confirmed track* q, $\mathbf{t}_{q,k}$ is not observed at time k, the most recent k_g time steps of trajectory positions are used as samples for GP regression. A *search boundary* $\mathbf{x}_k^S$ is constructed at time k for track q and predicted k_p time steps ahead using GP regression. After initialization, only the covariance of the *search boundary* is propagated based on the position $\mathbf{x}_{k+k_p}^S$, because the prediction model from GP regression becomes less confident as time goes.

E. Active Sensor Control

All the building blocks of the search and tracking algorithm (Figure 2) described so far focus on estimating the state of the targets. In this section, we focus on the complimentary problem of actively controlling the state of the sensor so as to improve the search and tracking process. A number of approaches has been proposed for active target tracking [33], target search [6], as well as joint search and tracking [24]. In this paper, we evaluate two simple strategies that are particularly suited to the underlying GM-PHD framework. Investigating better strategies with stronger performance guarantees is part of our ongoing work.

In GM-PHD, the mean of the Gaussian is a local maxima of the PHD (*i.e.*, most likely location to find a target in the local neighborhood), whereas the variance encodes the spatial uncertainty of the location of the targets. We evaluate two control strategies. (i) `nearest-gaussian`: drive to the nearest mean of all Gaussians in the mixture; and (ii) `largest-gaussian`: drive to the mean of the Gaussian with the largest covariance in the mixture. Once we reach the mean, the robot performs an outward spiral motion to search for the target in the corresponding *search boundary*.

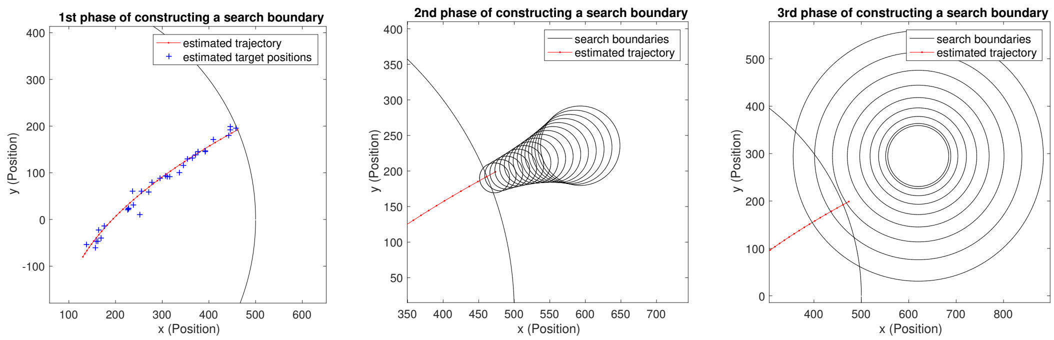


(a) Predicted trajectory with the interval along x–axis

(b) Predicted trajectory with the interval along y–axis

(c) Predicted *search boundaries* generated by covariance ellipses of GP regression

Fig. 3. Result of GP regression applied to a 2D trajectory sample.



(a) True trajectory with estimated target positions in the tracking region

(b) Predicted ellipses obtained from GP regression in the search region when $k_p = 15$

(c) Propagation of only covariance from a mean of ellipse at time $k+k_p$

Fig. 4. Process of the construction of a *search boundary*.

Intuitively, the `nearest-gaussian` strategy will track one or more targets for as long as possible, giving good tracking performance but poor search performance. On the other hand, the `largest-gaussian` strategy will equitably cover the search region giving good search performance but possibly poor tracking performance. We evaluate these two strategies through simulations. We hypothesize that a third strategy that switches between these two behaviors can be used to trade-off search and tracking objectives.

V. SIMULATION RESULTS

We implemented the proposed algorithm using an existing implementation for GM-PHD [21]. We present the results from two types of simulations. We first evaluate how well the GP regression performs as compared to a naive linear predictor for the motion model and predicts the *search boundary* of the targets. Then, we use the complete system to evaluate the two active sensor control strategies proposed.

A. Search Boundary Test

Figure 5 shows the effect of different values of hyperparameters on the *search boundary* for the x-coordinate of the estimated trajectory as a function of time. We use the squared-exponential kernel [27] which has three tunable hyperparameters (length-scale l, the signal variance σ_f^2, and the noise variance σ_n^2). All the boundaries were acquired from the 95% confidence interval. The black solid line is from the offline-trained values and other three lines show the interval when one of three hyperparameters is changed.

Figure 6 compares the GP regression model with a naive linear motion model of GM-PHD. To have a fair comparison, a linear motion is used to generate the true trajectory. We used the Hausdorff distance to compare the estimated trajectory with true trajectory. The linear estimator uses only one previous time step to make a prediction. We ran 30 tests and used the same test samples for both cases. Here sampling time steps were $k_g = 20$, and time steps of the prediction were $k_p = 20$, which start from time 0 in the figure. The training data for obtaining GP regression was different from the test samples. In all x and y-axis cases, both mean and standard deviation of the GP regression model are smaller than the linear model. This indicates that a *search boundary* generated by the linear model must be larger than the one from GP regression. It also turned out that the predicted trajectory from the linear model is sensitive to the last condition of a target before falling into the search region, which might lead to having a *search boundary* that already does not contain the true target.

B. Multitarget Search and Tracking

In this section, we report the results of applying our algorithm for representative search and tracking scenarios.

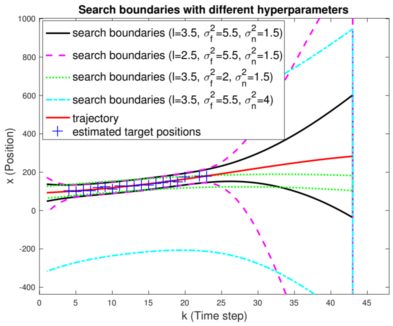
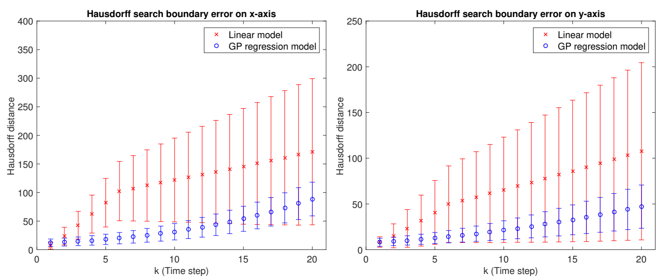


Fig. 5. Search boundaries along the x–coordinate of the target's trajectory with different hyperparameters (best viewed in color). The solid red curve is the true trajectory. The solid black lines mark the 95% confidence interval of the prediction derived using the offline learned hyperparameters.



(a) Prediction error along x–axis (b) Prediction error along y–axis

Fig. 6. Comparison of prediction error between the linear model of GM-PHD and the GP regression model. A target leaves the FoV at time 0 in this case after which the prediction is made. The true trajectories were generated from a linear motion model. We plot the Hausdorff error between the predicted and true trajectories.

The environment consisted of five search targets initially. The true number of targets is not known but an initial guess of approximate target positions is given in the form of Gaussian mixture distribution (Figure 7-left). The initial Gaussian mixture consisted of four components with different covariances. The robot has a circular FoV and can move with a constant velocity. At each time step, fifteen clutter measurements are generated, uniformly at random inside the FoV. We tested the two control strategies of Section IV-E using this initial setting and ran for 300 time steps each. When the sensor finds a target, it still approaches the target for certain amount of time steps to make sure that it becomes a *confirmed track*. On the other hand, when a target is not observed even at the mean of initial Gaussian distribution, the sensor takes a spiral motion to sweep the search boundary until it detects a target. If the sensor does not find a target within this region, the search target can be considered as a false positive.

Figure 7 shows the results for the two control strategies. The `nearest-gaussian` strategy ended up searching for the nearest target and then tracking it repeatedly. This is seen from Figure 8-(a) where the distance between the robot and target 1 always remains low, whereas the distance between the robot and other targets increases over time. On the other hand, the `largest-gaussian` strategy essentially visits all the means in the Gaussian mixture. Therefore, the number of tracking targets remains at zero for most of the time (Figure 8-(d)). As shown in Figure 8-(c,d), the true number of targets is five but estimated as four almost everywhere according to the wrong initial guess. However, the total estimated numbers of search and tracking targets fluctuate between three and five. This is the virtue of PHD as it estimates both the number of targets and target states.

VI. DISCUSSION AND CONCLUSIONS

Our main contribution in this paper is to extend the GM-PHD filter, initially proposed for the tracking problem [21], to allow for search and tracking with a limited FoV robot. Our second contribution was to incorporate non-linear target prediction using GP regression. The immediate future work is to incorporate better planning algorithms. In our previous work on particle PHD filters [4], we defined information-theoretic measures to control the position of the robots. Such approaches can directly be applied to the GM-PHD case. Another possible direction is to incorporate the *ridge-walking* algorithm [34] which plans a tour of level sets in the spatial distribution of the targets. However, this algorithm assumes that the targets are stationary and would thus need to be generalized to handle mobile target distributions.

REFERENCES

[1] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive bayesian search-and-tracking using coordinated uavs for lost targets," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006, pp. 2521–2526.

[2] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, 2003.

[3] B. Rao, H. F. Durrant-Whyte, and J. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[4] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots." in *International Symposium on Robotics Research*, 2015.

[5] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.

[6] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for invasive fish," *IEEE Robotics and Automation Magazine*, vol. 20, no. 3, pp. 33–41, 2013.

[7] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24 –39, Mar 2012.

[8] V. Isler, N. Noori, P. Plonski, A. Renzaglia, P. Tokekar, and J. Vander Hook, "Finding and tracking targets in the wild: Algorithms and field deployments," in *International Symposium on Safety, Security, and Rescue Robotics*, 2015, to appear.

[9] S. Radmard and E. A. Croft, "Active target search for high dimensional robotic systems," *Autonomous Robots*, pp. 1–18, 2015.

[10] B. Mobedi and G. Nejat, "3-d active sensing in time-critical urban search and rescue missions," *IEEE/ASME transactions on mechatronics*, vol. 17, no. 6, pp. 1111–1119, 2012.

[11] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2016.

[12] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[13] T. H. Chung and J. W. Burdick, "A decision-making framework for control strategies in probabilistic search," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4386–4393.

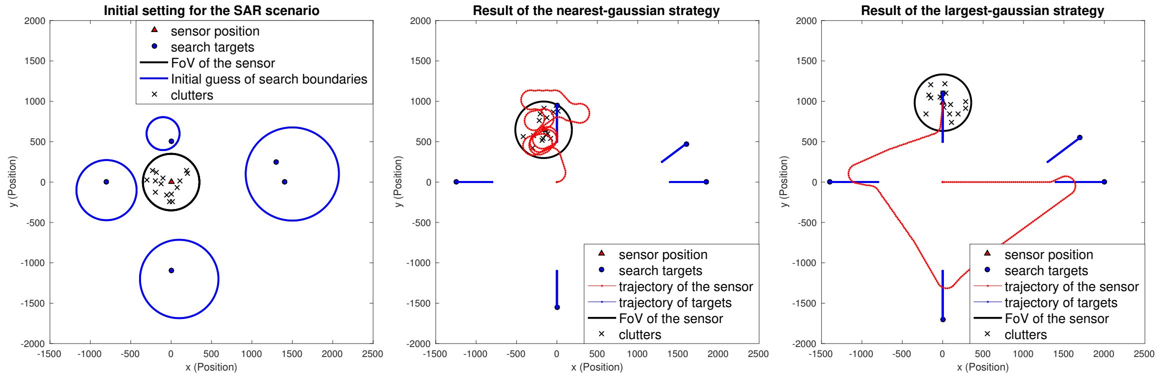
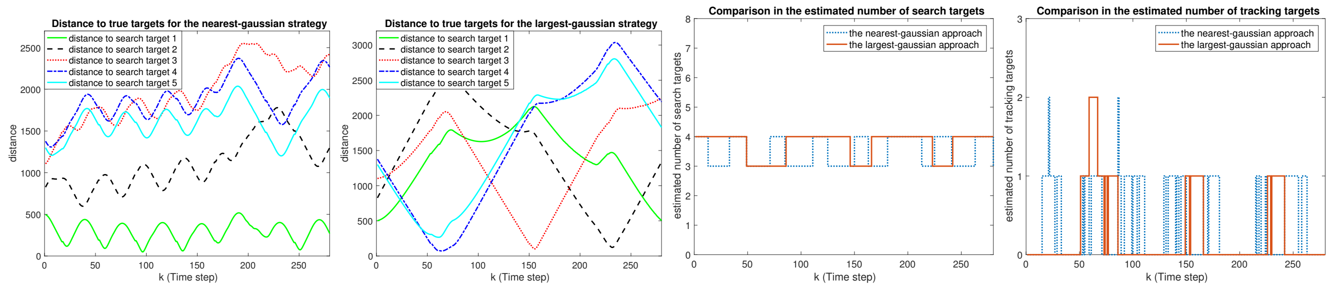


Fig. 7. **(left)** Initial configuration for the representative simulation. **(middle)** Final configuration for the `nearest-gaussian` strategy. **(right)** Final configuration for the `largest-gaussian` strategy. Please see the multi-media submission for an animation.



(a) (b) (c) (d)

Fig. 8. Distance to true targets for the (a) `nearest-gaussian` strategy and the (b) `largest-gaussian` strategy. The number of (c) search targets and (d) tracking targets for the two strategies.

[14] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 574–584, 2010.

[15] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[16] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, 2009.

[17] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[18] Y. Sung and W. Chung, "Hierarchical sample-based joint probabilistic data association filter for following human legs using a mobile robot in a cluttered environment," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 340–349, 2016.

[19] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Data association for semantic world modeling from partial views," *The International Journal of Robotics Research*, p. 0278364914559754, 2015.

[20] A. Ibarguren, I. Maurtua, M. Pérez, and B. Sierra, "Multiple target tracking based on particle filtering for safety in industrial robotic cells," *Robotics and Autonomous Systems*, vol. 72, pp. 105–113, 2015.

[21] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[22] R. P. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[23] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to bayesian slam," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, 2011.

[24] T. Furukawa, L. C. Mak, H. Durrant-Whyte, and R. Madhavan, "Autonomous bayesian search and tracking, and its experimental validation," *Advanced Robotics*, vol. 26, no. 5-6, pp. 461–485, 2012.

[25] P. Skoglar, U. Orguner, D. Törnqvist, and F. Gustafsson, "Road target search and tracking with gimballed vision sensor on an unmanned aerial vehicle," *Remote sensing*, vol. 4, no. 7, pp. 2076–2111, 2012.

[26] J. Tisdale, A. Ryan, Z. Kim, D. Tornqvist, and J. K. Hedrick, "A multiple uav system for vision-based search and localization," in *2008 American Control Conference*. IEEE, 2008, pp. 1985–1990.

[27] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[28] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A bayesian nonparametric approach to modeling motion patterns," *Autonomous Robots*, vol. 31, no. 4, pp. 383–400, 2011.

[29] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[30] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and electronic systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[31] K. Panta, D. E. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, 2009.

[32] D. E. Clark and J. Bell, "Multi-target state estimation and track continuity for the particle phd filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1441–1453, 2007.

[33] J. Vander Hook, P. Tokekar, and V. Isler, "Cautious greedy strategy for bearing-based active localization: Experiments and theoretical analysis," in *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1787–1792.

[34] C.-Y. Kim, D. Song, Y. Xu, J. Yi, and X. Wu, "Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1161–1173, 2014.