Distributed Multiple Hypothesis Tracker for Mobile Sensor Networks

Pujie Xin and Philip Dames

Temple University, Philadelphia, PA 19122, USA,
{pujie.xin,pdames}@temple.edu,
home page: https://sites.temple.edu/trail/

Abstract. This paper proposes a distributed estimation and control algorithm to allow a team of robots to search for and track an unknown number of targets. The number of targets in the area of interest varies over time as targets enter or leave, and there are many sources of sensing uncertainty, including false positive detections, false negative detections, and measurement noise. The robots use a novel distributed Multiple Hypothesis Tracker (MHT) to estimate both the number of targets and the states of each target. A key contribution is a new data association method that reallocates target tracks across the team. The distributed MHT is compared against another distributed multi-target tracker to test its utility for multi-robot, multi-target tracking.

Keywords: Distributed estimation, Multiple Hypothesis Tracker, Multi-target tracking

1 Introduction

The paper addresses the problem of multi-robot, multi-target tracking (MR-MTT), a canonical task in robotics, which includes problems such as search and rescue [21], surveillance [11], and mapping [10]. Additionally, we assume that the robots operate in a known area that contains an unknown number of targets (*e.g.,* the robots know the blueprint map of a building but not how many people are inside of it). The targets of interest can be stationary or dynamic, and they might enter or leave current environment. Thus the number of the targets varies with time. A solution to this problem contains two primary components: (I) a estimation algorithm to track the targets and (II) a control algorithm leads robots to search for new targets (exploration) while also keeping track of detected targets (exploitation).

Multiple Target Tracking (MTT) is a well studied problem, with the primary distinction between methods being how they address the problem of data association [30]. Common approaches include the Multiple Hypothesis Tracker (MHT) [2], Joint Probabilistic Data Association (JPDA) [13], Probability Hypothesis Density (PHD) filter [20], and Multi-target Multi-Bernoulli (MeMBer) Filter [33]. We base our solution on the track-oriented MHT (TOMHT) [16] as this offers a conceptually simple and computationally efficient solution.

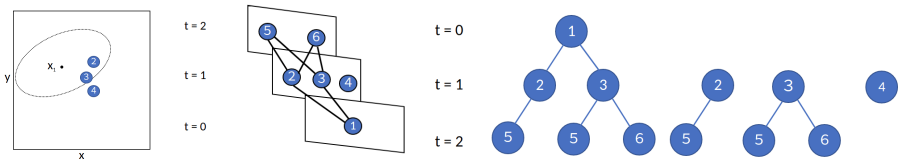
Active search with teams of robots is another well studied problem. The primary distinction in this space is the use of centralized [27, 34] versus distributed strategies [11, 19, 29]. One of the most successful distributed algorithms is Voronoi-based control [7], where the basic idea is to divide the search area with a Voronoi partition [12] and iterative move each robot to the centroid of its Voronoi cell, a process known as Lloyd's algorithm. In recent years, Multi-Agent Deep Reinforcement Learning (MADRL) offers a new method to generate search policies through repeated interactions between other agents and the environment in an attempt to maximize a reward function [15]. However, much of the work based on MADRL requires global information and uses a centralized architecture [25, 14]. Zhou et al. [35] propose a decentralized MADRL method using the maximum reciprocal reward to track cooperatively with UAV swarms. However, one downside of learning-based frameworks is that they often have trouble generalizing to unseen scenarios.

There are fewer works that specifically address the MR-MTT problem. One of the first solutions was Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) from Parker [22], which assigns robots to targets to maximize the time such that each target is observed by at least one robot. Related to CMOMMT, Pierson and Rus [23] proposed a distributed method for capturing multiple evaders with multiple pursuers in 2D and 3D environment by assigning pursuers to evaders and then using gradient descent. Pimenta et al. [24] propose Simultaneous Coverage and Tracking (SCAT) where they use Lloyd's algorithm to create a control law with guaranteed exponential convergence to a local minimum of the objective function. Our previous work [9] built upon SCAT by allowing the robots to learn the target distribution online rather than needing that information a priori. We did this by formulating a distributed PHD filter and using this as the importance weighting function in Lloyd's algorithm to drive target search. This paper will use the same control framework but formulate a new distributed MHT.

The primary contribution of this work is creating a new distributed TOMHT algorithm. Our approach leverages information about the relative locations of sensors (*i.e.,* robots) to decrease the memory and computational requirements relative to a naïve implementation. We then demonstrate the utility of this formulation to solve the MR-MTT problem by using the resulting target estimates to drive robot search. Our search strategy naturally and effectively drives the robots to follow previously detected targets and to explore unknown areas that may contain new targets. To demonstrate our result is correct and efficient, we compare our result to that of the PHD filter [9] through a series of simulated experiments with static and moving targets.

2 Background

In this section we provide the mathematical formulation of the standard Multiple Hypothesis Tracker (MHT) for multi-target tracking and of Lloyd's algorithm for distributed coverage control.



(a) Gating (b) Associated detections (c) Track trees

Fig. 1. Illustration of MHT. (a) Example of at $t = 1$, where observations 2 and 3 are in the gate and could be associated with the current track 1 while observation 4 is out of the gate and can not be associated to existing tracks. (b) 1-6 are observations at different time step. With the gating process, initial track hypothesis are built based on gating test. (c) The corresponding track trees. Each tree node is associated to an observation in (b).

2.1 MHT Definition

Let the set of objects be $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots\}$, where $\mathbf{x}_t^i$ is the state of the i^{th} object at time t. The MHT uses an extended Kalman filter to estimate the state of each object, so $\hat{\mathbf{x}}_t^i \sim \mathcal{N}(\boldsymbol{\mu}_t^i, \boldsymbol{\Sigma}_t^i)$. Since the number of objects is unknown, the MHT uses the concept of a *track* to represent a potential object. Therefore, the number of tracks represents the estimated number of objects within the search space and the state of each track represents the state of that individual object.

At each time step, a robot receives a set of measurements $\mathbf{Z}_t = \{\mathbf{z}_t^1, \mathbf{z}_t^2, \ldots\}$. The number of measurements depends upon the number of targets within the robot's field of view (FoV) as well as the probability of receiving a false negative p_{f_n} or false positive detection p_{f_p}. The MHT uses a two-step approach to solve the MTT, first solving the data association problem to match measurements to tracks and then using the associated measurements to update each track.

Data Association The MHT uses a gating method to determine if an individual measurement $\mathbf{z}_t^j$ should be associated with a track $\hat{\mathbf{x}}_t^i$:

$$d_t^2(\hat{\mathbf{x}}_t^i, \mathbf{z}_t^j) = (\hat{\boldsymbol{\mu}}_t^i - \mathbf{z}_t^j)^T (\boldsymbol{\Sigma}_t^i)^{-1} (\hat{\boldsymbol{\mu}}_t^i - \mathbf{z}_t^j) \leq \epsilon_d^2, \tag{1}$$

where $\boldsymbol{\mu}_t^i, \boldsymbol{\Sigma}_t^i$ are the mean and covariance of the estimated track $\hat{\mathbf{x}}_t^i$ and ϵ_d is the allowable distance threshold. Fig. 1a shows the gating process for one time step. This comparison is to determine the full set of potential associations between all measurements $\mathbf{Z}_t$ and all tracks $\hat{\mathbf{X}}_t$ at time t.

The TOMHT then considers this set of potential associations, as Fig. 1b shows. For example, if measurements $\mathbf{z}_t^2$ and $\mathbf{z}_t^3$ could be associated with a track $\hat{\mathbf{x}}_t^1$, then two copies of that track are propagated into the future, one using each hypothetical association, as Fig. 1c shows. All these distinct hypotheses for a track are maintained using a tree structure, where each branch is a unique series of associations and each node has a distinct mean and covariance. Additionally,

the TOMHT considers the possibility of starting a new tree using each measurement, which represents the possibility that a detection represents an entirely new object.

Track Scoring Based on the above structure, there will be many distinct hypotheses for an individual track based on different association histories. Therefore, the MHT uses a scoring system to determine the likelihood of individual tracks. This score can be used both to find the maximum likelihood track as well as to prune branches within the tree to avoid an exponential growth in the set of hypotheses over time. A standard way to formulate the score S uses the log likelihood ratio [3, 16]

$$S(A^i_{1:t}) = \ln \frac{p(\mathbf{z}^{k_{1:t}}_{1:t} \mid \mathbf{z}^{k_i}_i \subseteq \mathbf{Z}_i \, \forall i)}{p(\mathbf{z}^{k_{1:t}}_{1:t} \mid \mathbf{z}^{k_i}_i \subseteq \emptyset)} = \ln \frac{\prod_{i=1:t} p(\mathbf{z}^{k_i}_i \mid \mathbf{z}^{k_{i-1}}_{1:i-1}, \mathbf{z}^{k_i}_i \subseteq \mathbf{Z}_i)}{\prod_{i=1:t} p(\mathbf{z}^{k_i}_i \mid \mathbf{z}^{k_{i-1}}_{1:i-1}, \mathbf{z}^{k_i}_i \subseteq \emptyset)} \tag{2}$$

where $A^i_{1:t} = \{k_1, \ldots, k_t\}$ is a history of associations for track i, $p(\mathbf{z}^{k_i}_i \mid \mathbf{z}^{k_{i-1}}_{1:i-1}, \mathbf{z}^{k_i}_i \subseteq \mathbf{Z}_i) = \mathcal{N}(\mathbf{z}^{k_i}_i; \hat{\mathbf{x}}^{k_i}_i)$ represents the track being updated with a specific measurement and $p(\mathbf{z}^{k_i}_i \mid \mathbf{z}^{k_{i-1}}_{1:i-1}, \mathbf{z}^{k_i}_i \subseteq \emptyset) = 1/A_{\text{FoV}}$ represents a missed detection (*i.e.,* false negative) and A_{FoV} is the area of the sensor field of view. This score can be easily updated at each time

$$S(A^i_{1:t}) = S(A^i_{1:t-1}) + \begin{cases} \frac{p_{f_n}}{1-p_{f_p}}, & \mathbf{z}^{k_t}_t \subseteq \emptyset \\ \ln \frac{A_{\text{FoV}}}{2\pi} - \frac{1}{2} \ln |\boldsymbol{\Sigma}^t_{k_t}| + \frac{d^2}{2}, & \mathbf{z}^{k_t}_t \subseteq \mathbf{Z}_t \end{cases} \tag{3}$$

where $d(= 2)$ is the dimension of the space.

To get the maximum likelihood target set at time t, the TOMHT selects the best association history using the combination assignment problem:

$$\begin{aligned} &\max \sum_i S(A^i_{1:t}) \\ &\text{s.t. } A^i_{1:t} \cap A^j_{1:t} = \emptyset, \forall i \neq j \end{aligned} \tag{4}$$

In other words, we want to find the maximum total store such that each measurement is associated with at most one target. This optimization problem is solved in [16]. Then the N-scan pruning approach will trace back to $t - N$ and prune the subtrees which diverge from the global optimal hypothesis, which is the solution to (4).

2.2 Lloyd's Algorithm

Lloyds' algorithm [7] locally minimizes the function:

$$\mathcal{H}(q_1, q_2, ...q_R) = \int_E \min_{r \in \{1,\ldots,R\}} f(d(x, q_r))\phi(x)dx, \tag{5}$$

where $d(x, q)$ measures the distance between elements in E, $f(\cdot)$ is a monotonically increasing function, and $\phi(x)$ is a non-negative weighting function. $\phi(x)$ can be a time-invariant function in coverage problem and it can also be a time-variant function in tracking problem. A general choice is $f(x) = x^2$. The minimum value of the inside the integral is reached when we divide the environment using the Voronoi partition, $V_r = \{x \mid d(x, q_i) \leq d(x, q_j), \forall i \neq j\}$, where V_r are the Voronoi cells. The minimum with respect to robot positions is

$$(q^r)^* = \frac{\int_{V_r} x\phi(x)\, dx}{\int_{V_r} \phi(x)\, dx}, \tag{6}$$

This is a distributed control algorithm because as long as each robot knows the positions of its neighbors, a robot is capable of computing its Voronoi cell and move forward to its weighted centroid of Voronoi cell. Note that in practice, Voronoi neighbors can potentially be far apart. We make the assumption that the density of robots and/or the communication range of robots is sufficiently large to allow robots to communicate with all of their neighbors. The problem of formulating a consistent partition subject to communication range limits will the subject of future work.

This simple iterative, myopic control scheme is widely used in coverage problems. For example, Lee and Egerstedt [18] use Lloyd's algorithm to track a time-varying density function with a teams of robots. Bhattacharya et al. [1] expand the the search space to non-convex and non-Euclidean space. Suresh et al. [31] propose a method with low complexity and communication workload in distributed case using Lloyd's algorithm. Our previous work [9] repurposes this coverage controller to perform target tracking by choosing the estimated target density as the time-varying importance weighting function $\phi(x)$.

3 Distributed Multiple Hypothesis Tracker

The formulation above was for the standard single sensor case. However, two new considerations arise when there are multiple robots. First, multiple robots may observe the same target at the same time (and consequently may each have a local copy of a track for that target). Second, targets may leave the FoV of one robot and enter the FoV of another robot. So long as those two robots are within communication range (which is typically much larger than the sensing range), then the first robot can provide some information to the second robot to improve its initial estimation. This section outlines our novel formulation to address these issues.

Note that there are existing distributed MHT solutions for multi-sensor surveillance in radar systems [5, 6]. However, in that case the sensors are static and the topology of the communication network is fixed. In our case, the sensors are mobile, have a smaller FoV than that of radar, and the communication topology changes online. Our new formulation accounts for all of these effects.

3.1 Assumptions

We make three main assumptions. First, that robots know their pose. This is a strong assumption but robots operating in indoor environment with a high quality of priori map [26] or in outdoor environment with GPS receivers are able to navigate for long periods of time with minimal drift in their pose estimates. We also showed in our past work that sufficiently small drift has neglibigle effects on distributed target tracking accuracy [9]. Second, we assume that each robot is capable of communicating with its Voronoi neighbors and all the robots with overlapping sensor FoVs. This assumption was also discuss above in Sec. 2.2. Third, each robot has a unique id, which they will use to determine precedence to avoid conflicts.

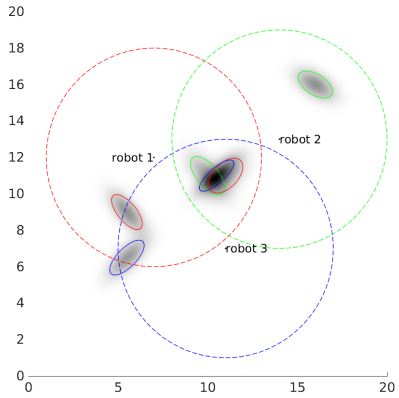


Fig. 2. Example of track fusion. There are three robots, whose FoV are shown by the dashed circles. Each robot has a unique color that is shared by its tracks. There are 4 targets, located at (10.5,11), (5.5,6.5), (5.5,9), and (16,16), and 6 tracks (colored ellipses), 2 from each robot. Here r_2 is the only robot with a nonempty set of private tracks, $\mathcal{P}^2 = (16, 16)$. All other tracks are in the union of multiple areas. Since r_3 has the largest ID, then $\mathcal{R}^3 = (5.5, 9), (10.5, 11)$ and $\mathcal{F}^3 = (10, 11), (11, 11), (5.5, 9)$, while $\mathcal{R}^1, \mathcal{R}^2, \mathcal{F}^1, \mathcal{F}^2 = \emptyset$. Then, r_3 calculates the KL-Divergence between tracks in $\mathcal{F}^3$ and $\mathcal{R}^3$ and fuse tracks with low KL-Divergence using Algorithm 1.

3.2 Track Exchange and Fusion

Similar to our past work creating a distributed PHD filter [9], our distributed MHT solution utilizes the Voronoi partition from Lloyd's algorithm to spatially distribute information across the team, *i.e.,* robot r maintains the tracks that are located within its Voronoi cell V_r. The key is to create a set of rules for data

exchange to ensure consistency across the team as cell boundaries shift due to robot motion and as targets move.

When robots are well spaced out with their FoVs not overlapping, then each robot can run its own separate MHT. However, when multiple robots view the same region then each may have its own separate track for the same target. These repeated tracks not only drive several robots track one target, which is a waste of system resource, but also increase false positive rate. Algorithm 1 outlines our approach, which is also illustrated in Fig. 2.

The process starts by inflating a robot's FoV to account for effects such as measurement noise (line 1), typically using the standard deviation of the measurement model 2σ. Robots then exchange these inflated regions to determine their local grouping $\mathcal{N}_F$ (line 2). Robots then exchange track estimates amongst these local clusters (lines 3-14), using the relative positions of the track and other robots as well as the IDs of each robot to sort each track into one of three categories: 1) private (*i.e.,* only in $\hat{\mathcal{F}}_r$), 2) fused (*i.e.,* in multiple regions but r has the highest ID amongst these), or 3) received (*i.e.,* in multiple regions but r is not the highest ID).

After this data exchange process, robot r measures the Kullback-Leibler divergence between all tracks in its fused and received sets $\mathcal{F}$ and $\mathcal{R}$ (lines 15-20). Robot r then uses the Munkres algorithm [17] to find the optimal assignment of tracks (line 21), where a valid assignment (*i.e.,* with cost less than $d_{\max}$) means that a fused and received track are the same object and the estimates are combined and added to the private list (lines 26-27). Unassigned tracks are added directly to the private list (lines 23-24 and 30-34).

3.3 Importance Weighting Function

Then we need to use the resulting target estimation from the MHT to drive exploration using Lloyd's algorithm. The key choice in Lloyd's algorithm is to set the importance weighting function $\phi(x)$. In our previous work [9], we used the PHD as the weighting function. We will take a similar approach, using

$$\phi(x) = v(x) + \sum_{i=1}^{n} \mathcal{N}(\hat{x}_i \mid \mu_i, \Sigma_i^2) \tag{7}$$

where $v(x)$ accounts for undetected targets (*i.e.,* driving exploration) and the right terms accounts for detected targets (*i.e.,* exploitation). We create a simple ad hoc rule to approximate the number of undetected targets

$$v(x) = \begin{cases} (1 - p_d(x))v(x) & x \in \text{FoV} \\ \max(v + \Delta v, \delta) & \text{else,} \end{cases} \tag{8}$$

where $p_d(x)$ is the probability of the robot detecting a target at x, Δv is the number of new targets potentially appearing per time step, and δ is a maximum target density to prevent "windup" in infrequently observed areas. The motivation behind this rule is to decrease the expected number of undetected

Algorithm 1 Track Exchange and Fusion for Robot r

$\hat{F}_r = F_r \oplus \mathcal{B}(2\sigma)$ ▷ Expand the FoV
$\mathcal{N}_F = \{j \neq r \mid \hat{F}_r \cap \hat{F}_j \neq \varnothing\}$
$\mathcal{P}, \mathcal{F}, \mathcal{R} = \emptyset$ ▷ Private, fused, and received track lists
for $t \in \mathcal{T}$ **do**
    $\mathcal{N}_t = \{j \in \mathcal{N}_F \mid t \in \hat{F}_j\}$
    if $\mathcal{N}_t$ is empty **then**
        $\mathcal{P} = \mathcal{P} \cup \{t\}$
    else if $r > \max_{j \in \mathcal{N}_t} j$ **then**
        $\mathcal{F} = \mathcal{F} \cup \{t\}$
    else
        Send t to robot $\max_{j \in \mathcal{N}_t} j$
    end if
end for
Add all tracks received from other robots to $\mathcal{R}$
for $t_i^{\mathcal{R}} \in \mathcal{R}$ **do** ▷ Create assignment distance matrix D
    for $t_j^{\mathcal{F}} \in \mathcal{F}$ **do**
        $D_{KL}(t_i^{\mathcal{R}} \| t_j^{\mathcal{F}}) = \frac{1}{2}\left(\|\mu_j^{\mathcal{F}} - \mu_i^{\mathcal{R}}\|^2_{(\Sigma_j^{\mathcal{F}})^{-1}} + \operatorname{tr}((\Sigma_j^{\mathcal{F}})^{-1}\Sigma_i^{\mathcal{R}}) - 2 + \ln \frac{\det \Sigma_j^{\mathcal{F}}}{\det \Sigma_i^{\mathcal{R}}} \right)$
        $d_{i,j} = \begin{cases} D_{KL}(t_i^{\mathcal{R}} \| t_j^{\mathcal{F}}), & D_{KL}(t_i^{\mathcal{R}} \| t_j^{\mathcal{F}}) < \epsilon_{\text{KDL}} \\ d_{\max}, & \text{else} \end{cases}$
    end for
end for
Find assignment $A : \mathcal{R} \to \mathcal{F}$
for $t_i^{\mathcal{R}} \in \mathcal{R}$ **do**
    if $A(i) = \emptyset$ **then** ▷ Track $t_i^{\mathcal{R}}$ not assigned
        $\mathcal{P} = \mathcal{P} \cup \{t_i^{\mathcal{R}}\}$
    else
        Make new track t with $\mu = \frac{\mu_i^{\mathcal{R}} + \mu_{A(i)}^{\mathcal{F}}}{2}$, $\Sigma = \frac{\Sigma_i^{\mathcal{R}} + \Sigma_{A(i)}^{\mathcal{F}}}{2}$
        $\mathcal{P} = \mathcal{P} \cup \{t\}$
    end if
end for
for $t_j^{\mathcal{F}} \in \mathcal{F}$ **do**
    if $(\nexists i$ s.t. $A(i) = j)$ AND $(d_{i,j} = d_{\max}, \forall i)$ **then** ▷ Track $t_j^{\mathcal{F}}$ not assigned
        $\mathcal{P} = \mathcal{P} \cup \{t_j^{\mathcal{F}}\}$
    end if
end for

targets for all points that are observed and increases the number of undetected targets outside of the field of view (where $\Delta v = 0$ when it is known that targets are stationary). We set $v(x) = \delta$ at the start of exploration. Typically, we set $\delta = 1/A$, where A is the area of the environment. This assumes there is only one target in the environment.

To make the problem tractable, we approximate the weighting function by a set of key points along a uniform grid in the environment, a common choice in robotics [32]. We then calculate the weight of each key point using (7) and use this to approximate the integrals in (6) by summations. We use the same algorithms as our previous work [9, Algorithm 1] to ensure the data is properly exchanged as Voronoi boundaries shift.

4 Results

In this section, we test our approach against our previous results using the distributed PHD filter [9], as this is the most closely related MR-MTT solution in the literature. First, we isolate the differences due to the tracker by using the same measurement and pose data from [9] and running this data through our new MHT. Then, we examine the benefits of the new weighting function (7) to show that unifying the estimation and control is helpful.

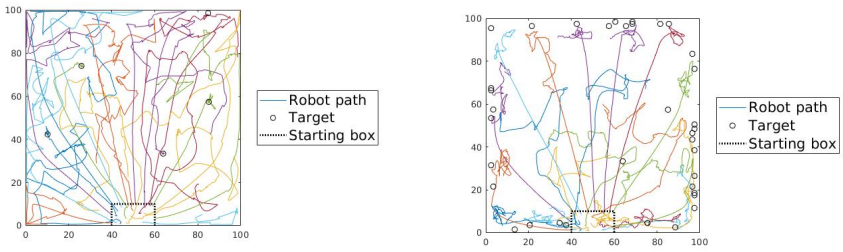
All trials take place in an open 100×100 m area. We run trials with different number of targets with the position of targets drawn uniformly from a 120×120 m area. Any targets initialized outside the map will be discarded. The robots begin at random locations within a small box at the bottom center of the map. The robot are holonomic with a maximum velocity of 2 m/s. Each robot is equipped with an onboard sensor with a circular FoV of radius 5 m, $p_{\text{fn}} = 1 - p_d = 0.2$, and $p_{\text{fp}} = 3.66 \cdot 10^{-3}$. Robots measure the relative position of a target $z \sim \mathcal{N}(x - q, 0.25I_2)$. The sensors collect measurements at 2 Hz. Note, these values do not represent any specific real-world situation, but could represent a team of drones equipped with downward facing cameras. For a real situation, one could follow the procedures outlined in our other work to carefully characterize the sensor models [8, 4].

4.1 Performance Metric

We measure tracking error using the Optimal SubPattern Assignment (OSPA) metric [28]. The error between sets X and Y, where $|X| = m \leq |Y| = n$ without loss of generality, is

$$d(X,Y) = \left(\frac{1}{n} \min_{\pi \in \Pi_n} \sum_{1}^{m} d_c(x_i, y_{\pi(i)})^p + c^p(n-m) \right)^{\frac{1}{p}} \tag{9}$$

where c is the cutoff distance (we use $c = 10$ m), $d_c(x,y) = \min(c, \|x-y\|)$, Π_n is the set of permutations of $\{1, 2, 3, ..., n\}$, and $p = 1$ is our choice for p-norm. Since the OSPA can fluctuate due to false negative and false positive detections,



(a) static targets (b) dynamic targets

Fig. 3. Robot trajectories. (a) shows the trajectories of robots during a single trail with 20 robots and 5 static targets. (b) shows the trajectories of robots during a single trail with 20 robots and dynamic targets

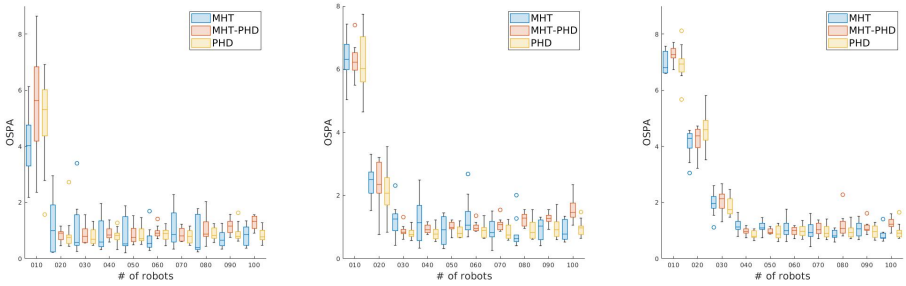


Fig. 4. Boxplots shows the median OSPA statistics over 10 runs for teams of 10-100 robots and 10,30,50 static targets for distributed MHT, MHT tracking based on control result in PHD and distributed PHD case respectively.

we calculate the median of the OSPA over the last 100 s to obtain a steady estimate and measure the variance of the OSPA over this same time window to measurement the stability of the tracker.

4.2 Stationary Targets

In the static case, the number of targets and their positions remain constant. We run a batch of trails with targets number 10, 30, and 50 respectively. The robots begin by distributing over the environment. Once a robot detects a target, it will typically continue to track that unless the track gets assigned to another robot, as Fig. 3a shows. As the robots explore the environment, $v(x)$ decreases in the detected area, thus the weighted centroid in one Voronoi cell keeps shifting away from regions without targets.

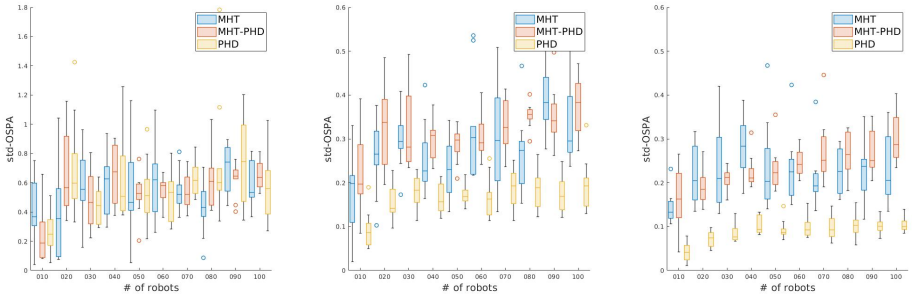


Fig. 5. Boxplots shows the median of standard deviation of OSPA statistics over 10 runs for teams of 10-100 robots and 10,30,50 static targets for distributed MHT, tracking results from MHT using detection results in PHD and distributed PHD case respectively.

Figure 4 shows the median OSPA over the last 100 s of a 550 s experiment (long enough to reach a steady state). We see that the median OSPA decreases significantly when the number of robots increases for all three methods, leveling out when the number of robots is near the number of targets. We see that in general, the performance is about the same using all three methods, with the mixed MHT-PHD case having a higher average OSPA as the control was not directly tied to the tracker. This is also born out in the standard deviation of the OSPA data in Fig. 5, where the MHT-PHD case tends to have the highest standard deviation. This also explains the increase in the standard deviation as the number of robots increases.

We also see that the MHT data tends to be between the mixed case and the PHD. This is likely due to the MHT making hard decisions about the presence of a target (*i.e.,* a track exists or is pruned) compared to the PHD, which uses a target density function the more gracefully degrades. We expect the MHT to perform better when there are fewer false negative detections, as repeated missed detections often cause tracks to be pruned.

4.3 Dynamic Targets

In the dynamic case, targets move around in the environment and the number of targets varies over time as targets enter or leave the map area. Thus we set up the birth rate $b(x)$ near the map boundary (∂E) to simulate the occurrence of new targets:

$$b(x) = \begin{cases} 5.26 \times 10^{-5} & \|x - \partial E\| \leq 5m \\ 0 & \text{else} \end{cases} \tag{10}$$
$$\triangle v(x) := \triangle v(x) + b(x)$$

Note, this birth model is incorporated into the $\triangle v$ term in (8). With this birth model, the total expected number of increasing targets is 1 per 10 s. Because of

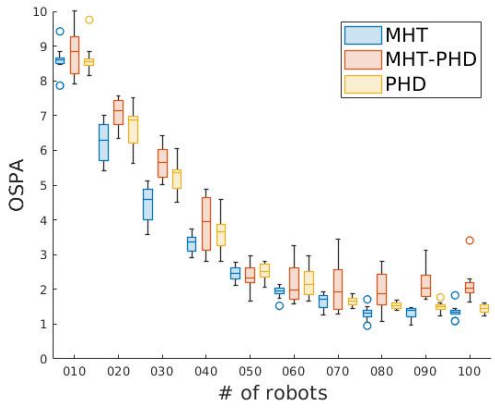


Fig. 6. Boxplots shows the median OSPA error statistics over 10 runs for teams of 10-100 robots and dynamic targets for distributed MHT, MHT tracking based on control result in PHD and distributed PHD case respectively.

this flux, the final number of targets is always similar regardless of the initial number. The emergent behavior in this case is different the static case, where some robots spread out along the map boundary due to detect new targets, as Fig. 3b shows. To get the steady state tracking performance, we extend the experiment to 800s and we evaluate the OSPA error over last 200s.

Figure 6 shows the median OSPA of the distributed MHT, MHT-PHD and PHD. We see that the OSPA decreases as the number of robots increases, and the variance of the OSPA has the same trend as the static case. The pure MHT estimation, namely, the MHT-PHD, still performs a little worse than the distributed PHD and keeps the largest variance among these three methods. The main potential reason is similar to that in static case, which is the control policy based on distributed PHD disturbs the tracks assignment in MHT estimation. However, unlike the static case, the MHT has a consistently lower OSPA value than the PHD data, especially in smaller teams. It also has a smaller variation in larger teams, meaning it provides a more consistent result.

We do not analyze the standard deviation of the OSPA in last 200s of each trail because the appearance and disappearance of the targets is the principle component of the standard deviation.

4.4 Discussion

Beyond the qualitative and quantitative tracking performance in the above scenario, there are a number of other factors to consider with MR-MTT algorithms.

Failure Cases The most challenging scenario for MTT is when targets are tightly clustered together (relative to the measurement covariance), as this is

when data association is most difficult. Due to their different approaches, the MHT and PHD filter will fail in different ways. The MHT makes multiple hard decisions about data association. Thus, tightly clustered targets will lead to a large branching factor at each time step, each of which may have a high likelihood of being true. This will cause the number of branches to be very large and difficult to prune. On the other hand, the PHD filter represents targets using one target density function. Thus, the PHD filter will have a single large peak in the density function, making it difficult to extract the precise location of any individual object within the cluster.

The other challenging scenario is when there are multiple missed detections in a row. In the MHT, this causes a track to be pruned prematurely, as there is typically a maximum timeout for a track to ensure that old branches get pruned. Increasing this timeout will make the MHT more robust to missed detections, but adds computations by increasing the number of tracks. The PHD filter fails in a different manner. In the PHD filter update equation [9, eq. (4)], the target density function decreases in areas where the robot expects to detect a target and increases in areas where the robot actually detects a target. Thus, repeated missed detections lower the density and make it difficult to recover quickly.

Adding multiple robots adds another challenge. Due to the design of our distributed MTT algorithms, when a target lies right at the boundary between two Voronoi cells then the ownership of that track will be ambiguous. Thus, two robots may repeatedly exchange ownership of a track simply due to measurement noise pushing the mean position across the Voronoi boundary. This will not affect tracking quality, but will increase the communication load.

Computational Complexity Analyzing the computational complexity of the MHT filter is challenging as the number of computations depends on the specific scenario under consideration. As mentioned above, the worst case scenario for the MHT is when all the targets are tightly clustered together (*i.e.,* within the gate radius of one another). In this case, the number of branches will grow exponentially over time, where the base of the exponential function is the number of targets in the cluster. On the other extreme, when targets are widely spaced then there will only be one branch per target, essentially just running an extended Kalman filter for each target in parallel.

The complexity of the PHD filter scales differently in the number of targets. As detailed in [9, eq. (4)], the PHD filter features a summation over the number of measurements (which is proportional to the number of targets). This superior scaling law relative to the MHT is a result of the choices made during the data association step of the MTT algorithm design.

The distributed MHT or PHD filter will generally scale better in the number of targets than a centralized MTT as the computations are distributed across the team, with each robot taking ownership of one portion of the full environment. The worst case would be if all targets in cell of a single robot, but this is almost guaranteed to never occur by construction of the control method, as robots are drawn towards areas of high target density.

Communication Load Like the complexity, the communication load is situationally dependent. There are three main stages to the distributed MHT or PHD filter algorithms, 1) Voronoi calculations, 2) environment reassignment, and 3) MTT update.

The Voronoi calculations are exactly the same between the distributed MHT and PHD filter and only require robots to exchange pose information with their neighbors, a low bandwidth operation.

The environment reassignment step in the MHT filter is where robots exchange tracks that have crossed the boundaries of Voronoi cells. The communication load depends upon the number of targets, the length of history, and the number of branches in the track tree. In the PHD filter, the robots exchange portions of the target density function (in our case, sets of weighted particles). Thus, with low target density the MHT will be more efficient but with high target density the PHD filter will be more efficient.

The MTT update step requires robots to exchange measurements if and only if they can see into the cell of another robot (*i.e.,* when the robot-robot distance is smaller than the field of view of a robot). In this case, the robots simply need to exchange measurement sets, which is a relatively low bandwidth operation as this is typically just a numerical array (*e.g.,* a list of range-bearing measurements). In the case of the PHD filter, the robots need to perform a second round of communication in order to compute a consistent normalization term in the PHD update step [9, eq. (5)].

5 Conclusion

In this paper, we propose a distributed algorithm to search for and track an unknown number of targets in a search area. There are two main components: 1) a novel, distributed MHT formulation for mobile robot teams and (2) a Voronoi-based control strategy. We compare the result of the distributed MHT to our previous work, which uses an alternative MTT solution, a distributed PHD filter. We found that both filters perform similarly well to solve the MR-MTT task. The PHD filter was slightly better in the case of static targets, yielding more accurate and consistent tracking performance. However, the MHT performed slightly better when the targets were dynamic, particularly when the size of the team is smaller. We also demonstrated the importance of tying the control directly to the tracking, with the MHT performing better on live data compared to running on the prior data collected by robots using the PHD filter.

Acknowledgement

This work was funded by NSF grants IIS-1830419 and CNS-2143312.

References

1. Bhattacharya, S., Ghrist, R., Kumar, V.: Multi-robot coverage and exploration on riemannian manifolds with boundaries. The International Journal of Robotics Research **33**(1), 113–137 (2014). DOI 10.1177/0278364913507324
2. Blackman, S.: Multiple hypothesis tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine **19**(1), 5–18 (2004). DOI 10.1109/MAES.2004.1263228
3. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House radar library. Artech House (1999)
4. Chen, J., Xie, Z., Dames, P.: The semantic PHD filter for multi-class target tracking: From theory to practice. Robotics and Autonomous Systems **149** (2022). DOI 10.1016/j.robot.2021.103947
5. Chong, C.Y., Mori, S., Reid, D.B.: Forty years of multiple hypothesis tracking - a review of key developments. In: 2018 21st International Conference on Information Fusion (FUSION), pp. 452–459 (2018). DOI 10.23919/ICIF.2018.8455386
6. Coraluppi, S., Carthel, C., Wu, C., Stevens, M., Douglas, J., Titi, G., Luettgen, M.: Distributed mht with active and passive sensors. In: 2015 18th International Conference on Information Fusion (Fusion), pp. 2065–2072 (2015)
7. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation **20**(2), 243–255 (2004). DOI 10.1109/TRA.2004.824698
8. Dames, P., Kumar, V.: Experimental characterization of a bearing-only sensor for use with the phd filter (2015)
9. Dames, P.M.: Distributed multi-target search and tracking using the PHD filter. Autonomous Robots **44**(3), 673–689 (2020). DOI 10.1007/s10514-019-09840-9
10. Deutsch, I., Liu, M., Siegwart, R.: A framework for multi-robot pose graph slam. In: 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 567–572 (2016). DOI 10.1109/RCAR.2016.7784092
11. Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M.W., Kosmatopoulos, E., Siegwart, R., Scaramuzza, D.: Optimal surveillance coverage for teams of micro aerial vehicles in gps-denied environments using onboard vision. Autonomous Robots **33**(1), 173–188 (2012). DOI 10.1007/s10514-012-9292-1
12. Du, Q., Faber, V., Gunzburger, M.: Centroidal voronoi tessellations: Applications and algorithms. SIAM Review **41**(4), 637–676 (1999). DOI 10.1137/S0036144599352836
13. Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE Journal of Oceanic Engineering **8**(3), 173–184 (1983). DOI 10.1109/JOE.1983.1145560
14. Goldhoorn, A., Garrell, A., Alquézar, R., Sanfeliu, A.: Searching and tracking people with cooperative mobile robots. Autonomous Robots **42**(4), 739–759 (2018). DOI 10.1007/s10514-017-9681-6
15. Gronauer, S., Diepold, K.: Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review **55**(2), 895–943 (2022). DOI 10.1007/s10462-021-09996-w
16. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: Proceedings of the IEEE international conference on computer vision, pp. 4696–4704 (2015)
17. Konstantinova, P., Udvarev, A., Semerdjiev, T.: A study of a target tracking algorithm using global nearest neighbor approach. In: Proceedings of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning, CompSysTech '03, p. 290–295 (2003). DOI 10.1145/973620.973668

18. Lee, S.G., Egerstedt, M.: Controlled coverage using time-varying density functions. IFAC Proceedings Volumes **46**(27), 220–226 (2013). DOI 10.3182/20130925-2-DE-4044.00030
19. Leonard, M.R., Zoubir, A.M.: Multi-target tracking in distributed sensor networks using particle phd filters. Signal Processing **159**, 130–146 (2019)
20. Mahler, R.: Multitarget bayes filtering via first-order multitarget moments. IEEE Transactions on Aerospace and Electronic Systems **39**(4), 1152–1178 (2003). DOI 10.1109/TAES.2003.1261119
21. Murphy, R.: Human-robot interaction in rescue robotics. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **34**(2), 138–153 (2004). DOI 10.1109/TSMCC.2004.826267
22. Parker, L.E.: Cooperative robotics for multi-target observation. Intelligent Automation & Soft Computing **5**(1), 5–19 (1999)
23. Pierson, A., Rus, D.: Distributed target tracking in cluttered environments with guaranteed collision avoidance. In: 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), pp. 83–89 (2017). DOI 10.1109/MRS.2017.8250935
24. Pimenta, L.C.A., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R.C., Pereira, G.A.S.: Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks, pp. 85–99. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-00312-7_6
25. Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., Wang, L.: Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. IEEE Access **7**, 146264–146272 (2019). DOI 10.1109/ACCESS.2019.2943253
26. Ribas, D., Ridao, P., Neira, J.: Localization with an a priori Map, pp. 47–75. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-14040-2_5
27. Robin, C., Lacroix, S.: Multi-robot target detection and tracking: taxonomy and survey. Autonomous Robots **40**(4), 729–760 (2016)
28. Schuhmacher, D., Vo, B.T., Vo, B.N.: A consistent metric for performance evaluation of multi-object filters. IEEE Transactions on Signal Processing **56**(8), 3447–3457 (2008). DOI 10.1109/TSP.2008.920469
29. Schwager, M., Dames, P., Rus, D., Kumar, V.: A multi-robot control policy for information gathering in the presence of unknown hazards. In: Springer Tracts in Advanced Robotics, pp. 455–472. Springer International Publishing (2016). DOI 10.1007/978-3-319-29363-9_26
30. Stone, L.D., Streit, R.L., Corwin, T.L., Bell, K.L.: Bayesian multiple target tracking. Artech House (2013)
31. Suresh, A.T., Yu, F.X., McMahan, H.B., Kumar, S.: Distributed mean estimation with limited communication. CoRR **abs/1611.00429** (2016)
32. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press, Cambridge, Mass. (2005)
33. Vo, B.T., Vo, B.N., Cantoni, A.: The cardinality balanced multi-target multi-bernoulli filter and its implementations. IEEE Transactions on Signal Processing **57**(2), 409–423 (2009). DOI 10.1109/TSP.2008.2007924
34. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems **10**(12), 399 (2013). DOI 10.5772/57313
35. Zhou, W., Li, J., Liu, Z., Shen, L.: Improving multi-target cooperative tracking guidance for uav swarms using multi-agent reinforcement learning. Chinese Journal of Aeronautics **35**(7), 100–112 (2022). DOI 10.1016/j.cja.2021.09.008