Trajectory PHD and CPHD filters

Ángel F. García-Fernández, Lennart Svensson

Abstract—This paper presents the probability hypothesis density filter (PHD) and the cardinality PHD (CPHD) filter for sets of trajectories, which are referred to as the trajectory PHD (TPHD) and trajectory CPHD (TCPHD) filters. Contrary to the PHD/CPHD filters, the TPHD/TCPHD filters are able to produce trajectory estimates from first principles. The TPHD filter is derived by recursively obtaining the best Poisson multitrajectory density approximation to the posterior density over the alive trajectories by minimising the Kullback-Leibler divergence. The TCPHD is derived in the same way but propagating an independent identically distributed (IID) cluster multitrajectory density approximation. We also propose the Gaussian mixture implementations of the TPHD and TCPHD recursions, the Gaussian mixture TPHD (GMTPHD) and the Gaussian mixture TCPHD (GMTCPHD), and the L-scan computationally efficient implementations, which only update the density of the trajectory states of the last \( L \) time steps.

Index Terms—Multitarget tracking, random finite sets, sets of trajectories, PHD, CPHD.

I. INTRODUCTION

The probability hypothesis density (PHD) and cardinality PHD (CPHD) filters are widely used random finite set (RFS) algorithms for multitarget filtering, which aims to estimate the state of the targets at the current time based on a sequence of measurements \([1]–[7]\). These filters have been successfully used in different applications such as multitarget tracking \([1]\), distributed multi-sensor fusion \([8], [9]\), robotics \([10], [11]\), computer vision \([12], [13]\), road mapping \([14]\) and sensor control \([15]\).

The PHD/CPHD filters fit into the assumed density filtering framework and propagate a certain type of multitrajectory density on the current set of targets through the prediction and update steps \([19]\). The PHD filter considers a Poisson multitarget density, in which the cardinality of the set is Poisson distributed and, for each cardinality, its elements are independent and identically distributed (IID). On the other hand, the CPHD filter considers an IID cluster multitarget density, in which the cardinality distribution of the set is arbitrary and, for each cardinality, its elements are IID. If the output of either prediction or update step is no longer Poisson/IID cluster, the PHD/CPHD filters obtain the best Poisson/IID cluster approximation by minimising the Kullback-Leibler divergence (KLD).

The most important benefit of the PHD/CPHD filters is their low computational burden, as they avoid the measurement-to-target association problem. However, their main drawbacks are their relatively low performance in some scenarios \([1], [17]\) and the fact that they do not build tracks, which denote sequences of target states that belong to the same target. The smoother versions of these filters \([1], [18], [19]\) do not solve these drawbacks. Despite the fact that the PHD/CPHD filters are unable to provide tracks in a mathematically rigorous way, several track building procedures have been proposed \([20]–[24]\). Another attempt to create tracks based on PHD/CPHD filters was proposed in \([25]\) by adding labels \([26], [27]\) to the target states. However, in the resulting labelled Poisson and labelled IID cluster densities, there is total confusion in the label-to-target association so they are not useful for track formation \([25]\ Sec. III.B). To solve this issue in \([25]\), apart from the unique labels, unique tags are added to the PHD components, as in \([22]\), and the original PHD/CPHD recursions are applied. However, in the considered posterior density, the tags are not part of the target state and are marginalised out. Therefore, the posterior is still distributed as labelled Poisson or labelled IID cluster and, theoretically, it does not have information to infer tracks.

In this paper, we address the intrinsic inability of PHD/CPHD filters to infer trajectories by developing PHD/CPHD filters that provide tracks from first principles, without adding labels or tags. We propose the trajectory PHD (TPHD) and trajectory CPHD (TCPHD) filters, which follow the same assumed density filtering scheme as the PHD/CPHD filters \([28]\) with a fundamental difference: instead of using a set of targets as the state variable, they use a set of trajectories \([29], [30]\).

The TPHD filter propagates a Poisson multitrajectory density on the space of sets of trajectories through the prediction and update steps, with a KLD minimisation after the update step. A diagram of the resulting Bayesian recursion is given in Figure 1. Similarly, the TCPHD filter propagates an IID cluster multitrajectory density and performs a KLD minimisation after the prediction and update steps. Due to the widespread use of PHD/CPHD filters, this paper covers an important gap in the literature, as we show how PHD/CPHD filtering can be endowed with the ability to infer trajectories in a rigorous way, and the KLD minimisation properties of the resulting filters. Apart from theoretically sound track formation, the proposed filters also have the advantage, compared to previous track building procedures used in PHD/CPHD filters, that they can update the information regarding past states of the trajectories.

In this paper, we also propose Gaussian mixture implementations of the TPHD/TCPHD filters, which follow the spirit of the Gaussian mixture PHD/CPHD filters \([3], [5]\). The resulting Gaussian mixture TPHD (GMTPHD) and TCPHD (GMTCPHD) filters build trajectories under a Poisson or IID cluster approximation, whose PHD is represented by
a Gaussian mixture. Additionally, we propose a version of the GMTPHD/GMTCPHD filters with lower computational burden called the \(L\)-scan GMTPHD/GMTCPHD filters. In practice, these filters only update the multitrajectory density of the trajectory states of the last \(L\) time instant leaving the rest unaltered, which is quite efficient for implementation. The theoretical foundation of the \(L\)-scan GMTPHD filter is also based on the assumed density filtering framework and KLD minimisations. Preliminary results of this paper covering the TPHD filter were presented in [25].

The remainder of the paper is organised as follows. Section II presents background material on sets of trajectories. In Section III, we introduce the Poisson and IID cluster multitrajectory densities and some of their properties. The TPHD and TCPHD filters are derived in Sections IV and V respectively. Their Gaussian mixture implementations are provided in Section VI. Simulation results are provided in Section VII. Finally, conclusions are drawn in Section VIII.

II. BACKGROUND

In this section, we describe some background material on multiple target tracking using sets of trajectories [30]. We review the considered variables, the set integral and cardinality distribution for sets of trajectories in Sections II-A, II-B and II-C respectively. Finally, we introduce the PHD for sets of trajectories in Section II-D.

A. Variables

A single target state \(x \in \mathbb{R}^{n_x}\) contains information of interest about the target, e.g., its position and velocity. A set of single target states \(X\) belongs to \(\mathcal{F}(\mathbb{R}^{n_x})\) where \(\mathcal{F}(\mathbb{R}^{n_x})\) denotes the set of all finite subsets of \(\mathbb{R}^{n_x}\). We are ultimately interested in estimating all target trajectories, where a trajectory consists of a sequence of target states that can start at any time step and end any time later on. Mathematically, a trajectory is represented as a variable \(X = (t, x^{1:i})\) where \(t\) is the initial time step of the trajectory, \(i\) is its length and \(x^{1:i} = (x^1, ..., x^i)\) denotes a sequence of length \(i\) that contains the target states at consecutive time steps of the trajectory.

We consider trajectories up to the current time step \(k\). As a trajectory \((t, x^{1:i})\) exists from time step \(t\) to \(t + i - 1\), variable \((t, i)\) belongs to the set \(I(k)\) is \(\{(t, i) : 0 \leq t \leq k \text{ and } 1 \leq i \leq k - t + 1\}\). A single trajectory \(X\) up to time step \(k\) therefore belongs to the space \(T(k) = \bigcup_{(t, i) \in I(k)} \{t\} \times \mathbb{R}^{n_x}\), where \(\bigcup\) stands for disjoint union, which is used to highlight that the sets are disjoint. Similarly to the set \(X\) of targets, we denote a set of trajectories up to time step \(k\) as \(\mathbf{X} \in \mathcal{F}(T(k))\).

Given a trajectory \(X = (t, x^{1:i})\), the set \(\tau^k(X)\), which can be empty, denotes the corresponding target state at a time step \(k\). Given a set \(\mathbf{X}\) of trajectories, the set \(\tau^k(\mathbf{X})\) of target states at time \(k\) is \(\tau^k(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} \tau^k(X)\).

B. Set integral

Given a real-valued function \(\pi(\cdot)\) on the single trajectory space \(T(k)\), its integral is

\[
\int \pi(X) \, dX = \sum_{(t, i) \in I(k)} \int \pi(t, x^{1:i}) \, dx^{1:i}.
\]

(1)

This integral goes through all possible start times, lengths and target states of the trajectory. Given a real-valued function \(\pi(\cdot)\) on the space \(\mathcal{F}(T(k))\) of sets of trajectories, its set integral is

\[
\int \pi(X) \, \delta X = \sum_{n=0}^{\infty} \frac{1}{n!} \int \pi(\{X_1, ..., X_n\}) \, dX_{1:n}
\]

(2)

where \(X_{1:n} = (X_1, ..., X_n)\). A function \(\pi(\cdot)\) is a multitrajectory density if \(\pi(\cdot) \geq 0\) and its set integral is one.

C. Cardinality distribution

Given a multitrajectory density \(\pi(\cdot)\), its cardinality distribution is

\[
\rho_\pi(n) = \frac{1}{n!} \int \pi(\{X_1, ..., X_n\}) \, dX_{1:n},
\]

(3)

which is analogous to the case where there is a set of targets.

D. Probability hypothesis density

The PHD [1] of a multitrajectory density \(\pi(\cdot)\) is

\[
D_\pi(X) = \int \pi(\{X\} \cup \mathbf{X}) \, \delta X.
\]

(4)

As in the PHD for RFS of targets, integrating the PHD in a region \(A \subseteq T(k)\) gives us the expected number of trajectories in this region [1] Eq. (4.76)]:

\[
\tilde{N}_A = \int_A D_\pi(X) \, dX = \sum_{(t, i) \in I(k)} \int 1_A(t, x^{1:i}) \, D_\pi(t, x^{1:i}) \, dx^{1:i}
\]

(5)

where \(1_A(\cdot)\) is the indicator function of a subset \(A\): \(1_A(z) = 1\) if \(z \in A\) and \(1_A(z) = 0\) otherwise. Therefore, the expected number of trajectories up to time step \(k\) is given by substituting \(A = T(k)\) into (5).

Example 1. Let us consider a multitrajectory density \(\nu(\cdot)\) with PHD

\[
D_\nu(1, x^{1}) = \mathcal{N}(x^1; 10, 1) + \mathcal{N}(x^1; 1000, 1)
\]

(6)

\[
D_\nu(1, x^{1:2}) = \mathcal{N}(x^{1:2}; (10, 10.1), \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix})
\]

(7)

d and \(D_\nu(X) = 0\) for \(X \neq (1, x^1)\) and \(X \neq (1, x^{1:2})\). \(\mathcal{N}(\cdot; m, P)\) is a Gaussian density with mean \(m\) and covariance
matrix $P$. The expected number of trajectories that start at time one with length 1 is given by substituting $A = \{1\} \times \mathbb{R}^{2x}$ into (5) so

$$\hat{N}_A = \int D_\nu (1, x^1) \, dx^1 = 2$$

The expected number of trajectories up to time step $k = 2$ is $\hat{N}_{T(k)} = 3$. □

### III. POISSON AND IID CLUSTER TRAJECTORY RFSs

In this section, we explain the Poisson and IID cluster trajectory RFSs.

#### A. Multitrajectory densities

1) **Poisson RFS**: For a Poisson RFS, the cardinality of the set is Poisson distributed and, for each cardinality, its elements are IID. A Poisson multitrajectory density $\nu (\cdot)$ has the form

$$\nu (\{X_1, \ldots, X_n\}) = e^{-\lambda_\nu} \lambda_\nu^n n! \int \hat{\nu} (X_j)$$

where $\hat{\nu} (\cdot)$ is a single trajectory density, which implies

$$\int \hat{\nu} (X) \, dX = 1,$$

and $\lambda_\nu \geq 0$. A Poisson multitrajectory density is characterised by either its PHD $D_\nu (X) = \lambda_\nu \hat{\nu} (X)$ or by $\lambda_\nu$ and $\hat{\nu} (\cdot) [1]$. As a result, using (5), the expected number of trajectories is $\hat{N}_{T(k)} = \lambda_\nu$. Further, its cardinality distribution is given by

$$\rho_\nu (n) = \frac{1}{n!} \int \nu (\{X_1, \ldots, X_n\}) \, dX_{1:n} = \frac{1}{n!} e^{-\lambda_\nu} \lambda_\nu^n.$$  

(10)

**Example 2.** We consider a Poisson RFS with the PHD of Example 1. Using (10), its cardinality distribution is Poisson with $\lambda_\nu = 3$ and, therefore, its single trajectory density is $\hat{\nu} (X) = D_\nu (X)/3$. □

2) **IID cluster RFS**: For an IID cluster RFS with density $\nu (\cdot)$, the cardinality is distributed according to the probability mass function $\rho_\nu (\cdot)$ and, for each cardinality, its elements are IID according to a single trajectory density $\hat{\nu} (\cdot)$. The resulting multitrajectory density is

$$\nu (\{X_1, \ldots, X_n\}) = \rho_\nu (n) n! \prod_{j=1}^{n} \hat{\nu} (X_j).$$

(11)

As $\hat{\nu} (\cdot)$ is a single trajectory density, it meets [9]. The PHD of (11) is given by [11]

$$D_\nu (x) = \hat{\nu} (x) \sum_{n=0}^{\infty} n \rho_\nu (n)$$

(12)

where the second factor corresponds to the expected number of targets. An IID cluster density can be characterised either by $\rho_\nu (\cdot)$ and $\hat{\nu} (\cdot)$, or by $\rho_\nu (\cdot)$ and $D_\nu (\cdot)$.

#### B. Sampling

In this subsection, we indicate how to draw samples from an IID cluster trajectory RFS, which includes the Poisson trajectory RFS as a particular case. Given a single trajectory density $\hat{\nu} (\cdot)$, the probability that the trajectory starts at time $t$ and has duration $i$ is

$$P_\nu (t, i) = \int \hat{\nu} (t, x^{1:i}) \, dx^{1:i}.$$  

(13)

Given the start time $t$ and duration $i$, the density of the target states of the trajectory is

$$\bar{\nu} \left( x^{1:i} \mid t, i \right) \equiv \frac{\hat{\nu} (t, x^{1:i})}{P_\nu (t, i)},$$

(14)

We can draw samples from an IID cluster trajectory RFS using (13) and (14), as indicated in Algorithm 1.

**Algorithm 1** Sampling from an IID cluster multitrajectory density

**Input:** IID cluster multitrajectory density $\nu (\cdot)$.

**Output:** Sample $X \sim \nu (\cdot)$.

- Set $X = \emptyset$ and sample $n \sim \rho_\nu (\cdot)$.
  - for $j = 1$ to $n$ do
    - Sample $(t, i) \sim P_\nu (\cdot)$ and $x^{1:i} \sim \hat{\nu} (\cdot) \mid t, i$, see (13) and (14).
    - Set $X \leftarrow X \cup \{ (t, x^{1:i}) \}$.
  - end for

#### C. KLD minimisation

In this subsection, we provide two KLD minimisation theorems for Poisson and IID cluster multitrajectory densities, which will be used to derive the trajectory PHD/CPHD filters. The KLD $D (\pi \| \nu)$ between multitrajectory densities $\pi (\cdot)$ and $\nu (\cdot)$ is given by [11]

$$D (\pi \| \nu) = \int \pi (X) \log \frac{\pi (X)}{\nu (X)} \, dX.$$  

(15)

Then, the following theorems hold:

**Theorem 3.** Given a multitrajectory density $\pi (\cdot)$, the Poisson multitrajectory density $\nu (\cdot)$ that minimises the KLD $D (\pi \| \nu)$ is characterised by the PHD $D_\nu (\cdot) = D_\pi (\cdot)$.

**Theorem 4.** Given a multitrajectory density $\pi (\cdot)$, the IID cluster multitrajectory density $\nu (\cdot)$ that minimises the KLD $D (\pi \| \nu)$
Theorem 3 is proved in Appendix A in [28]. The analogous theorem for sets of targets was proved in [31]. Theorem 4 is proved in Appendix A in the supplementary material. The analogous theorem for sets of targets was proved in [16], [32]. It should be noted that, as a Poisson RFS is a special type of IID cluster RFS, the best fitting IID cluster RFS always has a lower or equal KLD than the best fitting Poisson RFS.

D. Inference only on alive trajectories

In this section, we explain why the TPHD and TCPHD filters are mainly useful to approximate the posterior multitrajecory density over alive trajectories, but not the posterior over all trajectories, which also include dead trajectories. This serves as a motivation to present the TPHD and TCPHD filters for tracking only the alive trajectories in the next sections.

Let us first explain why the TPHD filter, which considers a Poisson approximation, is only useful for the alive trajectories [28, Sec. V.B], though it was derived in [28] for dead and alive trajectories. In the prediction step, the part of the PHD that represents a trajectory that dies at the current time step is multiplied by the probability of death (one minus the probability of survival) [28, Thm. 5], which is usually low. As time goes on, the part of the PHD that represents dead trajectories never changes. As a result, even if a trajectory exists with a very high probability at some point in time, once it dies, the TPHD filter over all trajectories indicates that it existed with a very low probability. Therefore, the TPHD does not contain accurate information about dead trajectories, though it does contain useful information about alive trajectories.

In the following, we argue with an example why the TCPHD filter, which considers an IID cluster approximation, should only consider alive trajectories, as the TPHD filter.

Example 5. Let us consider that the posterior \( \pi^k (\cdot) \) over the set of trajectories at time \( k \) has \( m \) trajectories with probability 1 so \( \rho_\pi (m) = 1 \). In addition, \( \pi^k (\cdot) \) indicates that there are \( m_d \) dead trajectories with independent (single trajectory) densities \( d_1 (\cdot), \ldots, d_{m_d}(\cdot) \), and \( m_a \) alive trajectories with independent densities \( a_1 (\cdot), \ldots, a_{m_a}(\cdot) \), where \( m_d + m_a = m \). Note that we can obtain this kind of true posterior, without TPHD/TCPHD approximations, if the probability of detection is one, there is no clutter, targets are born independently and they are far from each other at all time steps. In Appendix B (see supplementary material), we compute the best IID cluster density approximation \( \nu^k (\cdot) \) to \( \pi^k (\cdot) \) using Theorem 4 and show that the cardinality distribution of the alive targets in \( \nu^k (\cdot) \) is

\[
\rho_a (n) = \binom{m}{n} \left( \frac{m_a}{m} \right)^n \left( 1 - \frac{m_a}{m} \right)^{m-n},
\]

where \( n \in \{ 0, 1, \ldots, m \} \). As the filtering recursion continues, the total number \( m \) of trajectories can only increase. On the contrary, \( m_a \) does not necessarily increase so after a sufficiently long time \( \frac{m_a}{m} \) may become very small. Then, using the Poisson limit theorem, the cardinality distribution of the alive targets can be approximated as Poisson with parameter \( m_a \) [33]. Therefore, even in this simple example in which the cardinality of the alive targets is known, the best IID cluster approximation of the whole trajectory posterior approximates the cardinality of the alive targets as a Poisson distribution.

The conclusion of the previous example is that, in the long run, an IID cluster RFS is not necessarily better than a Poisson RFS, both considered over all trajectories, to approximate the cardinality distribution of the alive targets. In most applications, the cardinality of the alive trajectories is considerably more important than the cardinality of the total number of trajectories. In this paper, we therefore focus on an IID cluster approximation of the alive trajectories to develop the TCPHD filter. This implies that the TCPHD filter has an arbitrary cardinality distribution for the alive targets, as the CPHD filter.

IV. Trajectory PHD filter

In this section, we derive the TPHD filter for tracking the alive targets. In Section IV-A, we present the Bayesian filtering recursion for sets of trajectories. The prediction and update steps of the TPHD filter are given in Sections IV-B and IV-C respectively.

A. Bayesian filtering recursion

The posterior multitrajecory density \( \pi^k (\cdot) \) at time \( k \), which denotes the density of set of trajectories present at time \( k \) given all measurements up to time \( k \), is calculated via the prediction and update steps:

\[
\omega^k (X) = \int f (X|Y) \pi^{k-1} (Y) \delta Y
\]

\[
\pi^k (X) = \frac{\ell^k (z^k|\pi^k (X)) \omega^k (X)}{\ell^k (z^k)}
\]

where \( f (\cdot|\cdot) \) is the transition density, \( \omega^k (\cdot) \) is the predicted density at time \( k \), \( z^k \) is the set of measurements at time \( k \), \( \ell^k (\cdot|\pi^k (X)) \) is the density of the measurements given the current RFS of targets and

\[
\ell^k (z^k) = \int \ell^k (z^k|\pi^k (X)) \omega^k (X) \delta X
\]

is the density of the measurements given the predicted density \( \omega^k (\cdot) \). The predicted density at time \( k \) is the density of the set of trajectories present at time step \( k \) given the measurements up to time step \( k - 1 \). As we only take into account the present trajectories, the only term that changes in (17)-(18) with respect to considering all trajectories is \( f (\cdot|\cdot) \), see [34, Sec. IV-A] for a detailed explanation. The description of these models will be given in Sections IV-B and IV-C.

B. Prediction

We make the following assumptions in the prediction step:

- P1 Given the current multitarget state \( x \), each target \( x \in x \) survives with probability \( p_S (x) \) and moves to a new state with a transition density \( g (\cdot|x) \), or dies with probability \( 1 - p_S (x) \).
• P2 The multitarget state at the next time step is the union of the surviving targets and new targets, which are born independently with a Poisson multitarget density $\beta_\tau(\cdot)$.

• P3 The multitrajectory density $\pi_k(\cdot)$ of the trajectories present at time $k$ represents a Poisson RFS.

Note that we use subindex $\tau$ in densities on RFS of targets, as in $\beta_\tau(\cdot)$. Let $\mathbb{N}_k = \{1, \ldots, k\}$. Then, the relation between predicted PHD at time $k$ and the PHD of the posterior at time $k - 1$ is given by the following theorem.

**Theorem 6 (TPHD filter prediction).** Under Assumptions P1-P3, the predicted PHD $D_{\omega^k}(\cdot)$ of the trajectories present at time $k$ is

$$D_{\omega^k}(X) = D_{\xi^k}(X) + D_{\beta^k}(X) \quad (19)$$

where

$$D_{\beta^k}(t, x^{1:i}) = D_{\beta_\tau}(x^1) 1_{\mathbb{N}_k}(t)$$

$$D_{\xi^k}(t, x^{1:i}) = p_S(x^{i-1}) g(x^i| x^{i-1}) \times D_{\pi_{k-1}}(t, x^{1:i-1}) 1_{\mathbb{N}_{k-1}}(t)$$

if $t + i - 1 = k$ or zero otherwise.

This theorem is proved in [28] for a more general case in which dead trajectories are considered. As mentioned in Section II-D in this paper, we only present the results for alive trajectories, as the results are mainly useful in this case. The predicted PHD is the sum of the PHD $D_{\beta^k}(\cdot)$ of the trajectories born at time step $k$ and the PHD $D_{\xi^k}(\cdot)$ of the surviving trajectories. The end time of trajectory $(t, x^{1:i})$ is $t + i - 1$ so $D_{\omega^k}(t, x^{1:i})$ is zero if $t + i - 1 \neq k$.

For the surviving trajectories, we multiply the PHD by the transition density and the survival probability. Note that the provided PHD characterises the Poisson RFS that represents the predicted density.

**C. Update**

We make the following assumptions in the update step:

• U1 For a given multi-target state $x$ at time $k$, each target state $x \in x$ is either detected with probability $p_D(x)$ and generates one measurement with density $l(\cdot|x)$, or missed with probability $1 - p_D(x)$.

• U2 The measurement $z^k$ is the union of the target-generated measurements and Poisson clutter with density $c(\cdot)$.

• U3 The multitrajectory density $\omega^k(\cdot)$ represents a Poisson RFS.

Let $\Xi_n, n \in \mathbb{N}$ denote the set that contains all the vectors $\sigma = (\sigma_1, \ldots, \sigma_n)$ that indicate associations of $n$ measurements to $n$ targets, which can be either detected or undetected. If $\sigma \in \Xi_n, n \in \mathbb{N}$, $\sigma_i = j \in \{1, \ldots, n\}$ indicates that measurement $j$ is associated with target $i$ and $\sigma_i = 0$ indicates that target $i$ has not been detected. Under Assumptions U1 and U2, which define the standard measurement model, the density of the measurement given the state is [1] Eq. (7.21)]

$$\ell^k \left( \{z_1, \ldots, z_n\} \mid \{x_1, \ldots, x_n\} \right) = e^{-\lambda_c} \prod_{i=1}^{n_1} \lambda_c \tilde{c}(z_i) \prod_{i=1}^{n} \left(1 - p_D(x_i)\right) \prod_{\sigma \in \Xi_n, n \sigma_i > 0} p_D(x_i) l(\sigma, x_i) \left(1 - p_D(x_i)\right) \lambda_c \tilde{c}(\sigma), \quad (20)$$

where $\lambda_c$ and $\tilde{c}(\cdot)$ characterise $c(\cdot)$, see [8].

Let $L_{z^k}(\cdot)$ denote the PHD filter pseudolikelihood function, which is given by [1 Sec. 8.4.3]

$$L_{z^k}(x) = 1 - p_D(x) + p_D(x) \times \sum_{z \in \Xi^k} \lambda_c \tilde{c}(z) + \int p_D(y) l(z|y) D_{\omega^k}(y) dy$$

with $D_{\omega^k}(\cdot)$ representing the PHD of the targets at time $k$ of density $\omega^k(\cdot)$, which is given by [28]

$$D_{\omega^k}(y) = \int_{t=1}^{k} D_{\omega^k}(t, x^{1:k-t}, y) dx^{1:k-t}. \quad (21)$$

Then, the TPHD filter update step is given by the following theorem:

**Theorem 7 (TPHD filter update).** Under Assumptions U1-U3, the updated PHD $D_{\omega^k}(\cdot)$ at time $k$ is

$$D_{\omega^k}(t, x^{1:i}) = D_{\omega^k}(t, x^{1:i}) \circ L_{z^k}(x) \quad (22)$$

if $t + i - 1 \leq k$ or zero, otherwise.

This theorem is proved in [28] for a more general case in which dead trajectories are included. It should be noted that Bayes update (18) uses a likelihood (20) that involves a summation over all target to measurements associations in the multitarget space. In contrast, the TPHD filter update is similar to the PHD filter update in the sense that it uses a pseudolikelihood function $L_{z^k}(\cdot)$ which is defined on the single target space and only involves associations between a single target and the measurements.

**V. TRAJECTORY CPHD FILTER**

In this section we present the trajectory CPHD (TCPHD) filter for tracking the alive targets. The TCPHD filter works within the assumed density filtering framework by approximating the multitrajectory densities of the alive trajectories as IID cluster densities throughout the filtering recursion, see Figure 2.

Prior to deriving the TCPHD filter, we provide some notation. Given two sequences $a(n)$ and $b(n)$, $n \in \mathbb{N} \cup \{0\}$, we denote

$$\langle a, b \rangle = \sum_{n=0}^{\infty} a(n) b(n).$$

Given a set $Z$, the elementary symmetric function of order $j$ is [5]

$$e_j(Z) = \sum_{\sum_{\xi \in Z} |\xi| = j} \prod_{\xi \in Z} (\xi)$$

with $e_0(Z) = 1$ by convention. We also use \ to denote set subtraction.
distribution of the targets present at time $t$. Under Assumption U5, it is direct to obtain, that the distribution of the present targets at the current time, which is performed as indicated by Theorem 4. We first apply Bayes’ rule, see (18), followed by a KLD minimisation, which is performed as indicated in Figure 2. The TCPHD update consists of past states of the trajectories in the PHD to keep trajectory information, while the CPHD filter does.

As indicated in Figure 2, the TCPHD update consists of applying Bayes’ rule, see (18), followed by a KLD minimisation, which is performed as indicated by Theorem 4. We first consider the distribution of the present targets at the current time. Under Assumption U5, it is direct to obtain, that the distribution of the targets present at time $k$ is also an IID cluster with the cardinality distribution $\rho_{\pi^k} (\cdot)$ and PHD (21). The resulting TCPHD filter update is given in the following theorem.

**Theorem 9** (TCPHD filter update). Under Assumptions U1, U4 and U5, the cardinality distribution and the PHD of the posterior at time $k$ are

$$
\rho_{\pi^k} (n) = \sum_{m=0}^{\infty} \rho_{\pi^k} (m) \sum_{j=0}^{\infty} \binom{n}{j} \rho_{\pi^{k-1}} (n) \times \left[ \int (1 - p_S (x)) D_{\pi^{k-1}} (x) dx \right]^{n-j} \times \left[ \int D_{\pi^{k-1}} (x) dx \right]^{-j} \times \left[ \int p_S (x) D_{\pi^{k-1}} (x) dx \right]^{j} \tag{23}
$$

where $D_{\pi^{k-1}} (\cdot)$ is the PHD of the targets at time $k - 1$ according to $\pi^{k-1} (\cdot)$, which is calculated as in (21).

Theorem 9 is proved in Appendix C (see supplementary material). The update step of the TCPHD filter is equivalent to the CPHD filter, with the main difference that the updated PHD contains information about previous states of the trajectories.

**VI. GAUSSIAN MIXTURE IMPLEMENTATIONS**

In this section, we present the Gaussian mixture implementations of the TPHD and TCPHD filters. We use the notation

$$
\mathcal{N} (t, x^{1:i}; t^k, m^k, P^k) = \begin{cases} 
\mathcal{N} (x^{1:i}; m^k, P^k) & t = t^k, i = i^k \\
0 & \text{otherwise}
\end{cases} \tag{27}
$$

where $i^k = \dim (m^k) / n_x$. Equation (27) represents a single trajectory Gaussian density with start time $t^k$, duration $i^k$, mean $m^k \in \mathbb{R}^{ik \times n_x}$ and covariance matrix $P^k \in \mathbb{R}^{ik \times n_x \times ik \times n_x}$ evaluated at $(t, x^{1:i})$. We use $\otimes$ to indicate the Kronecker product and $0_{m,n}$ is the $m \times n$ zero matrix.

We make the additional assumptions

- A1 The probabilities $p_S$ and $p_D$ are constants.
- A2 $g (x^1 | x^{i-1}) = N (x^1; Fx^{i-1}, Q)$.
- A3 $l (z|x) = N (z; Hx, R)$.
- A4 The PHD of the birth density $\beta^k (\cdot)$ is

$$
D_{\beta^k} (X) = \sum_{j=1}^{J^k} w^k_{\beta,j} \mathcal{N} (X; k, m^k_{\beta,j}, P^k_{\beta,j}) \tag{28}
$$
where $J^k_j \in \mathbb{N}$ is the number of components, $w_{\omega,j}^k$ is the weight of the $j$th component, $m_{\omega,j}^k \in \mathbb{R}^{n_x}$ its mean and $P_{\omega,j}^k \in \mathbb{R}^{n_x \times n_x}$ its covariance matrix.

It should be noted that the models provided by A1-A4 could be time varying but time is omitted for notational convenience. In the rest of this section, we present the Gaussian mixture implementations of the TPHD and TCPHD filters in Sections VI-A and VI-B respectively. The L-scan versions of the filters and trajectory estimation are addressed in Sections VI-C and VI-D. Finally, a discussion is provided in Section VI-E.

A. Gaussian mixture TPHD filter

The recursion of the GMTPHD filter is similar to the GMPHD filter [3], but it keeps trajectory information. Under Assumptions A1-A4, P1-P3 and U1-U3, we can calculate the TPHD filter in closed form giving rise to the GMTPHD filter, whose prediction and update steps are provided in the following propositions.

Proposition 10 (GMTPHD filter prediction). Assume $\pi^k(\cdot)$ has a PHD

$$D_{\pi^k}(X) = \sum_{j=1}^{J^k} w_j^k \mathcal{N}(X; t_j^k, m_j^k, P_j^k)$$

where $t_j^k + i_j^k - 1 = k$ with $i_j^k = \text{dim}(m_j^k)/n_x$. Then, the PHD of $\omega^{k+1}(\cdot)$ is

$$D_{\omega^{k+1}}(X) = D_{\beta^{k+1}}(X) + ps \sum_{j=1}^{J^k} w_j^k \mathcal{N}(X; t_j^k, m_{\omega,j}^{k+1}, P_{\omega,j}^{k+1})$$

where

$$m_{\omega,j}^{k+1} = \left[ \begin{array}{c} (m_j^k)^T, (F_j m_j^k)^T \\ 0_{1, i^k_j-1}, 1 \end{array} \right] \otimes F$$

$$P_{\omega,j}^{k+1} = \left[ \begin{array}{c} P_j^k F_j^T \\ F_j P_j^k F_j^T + Q \end{array} \right].$$

Proposition 10 is a consequence of Theorem 6 and conventional properties of Gaussian densities. Compared to the GMPHD filter prediction, the main difference is that previous states are not integrated out.

Proposition 11 (GMTPHD filter update). Assume $\omega^k(\cdot)$ has a PHD

$$D_{\omega^k}(X) = \sum_{j=1}^{J^k} w_{\omega,j}^k \mathcal{N}(X; t_{\omega,j}^k, m_{\omega,j}^k, P_{\omega,j}^k).$$

Then, the PHD of $\pi^k(\cdot)$ is

$$D_{\pi^k}(X) = (1 - p_D) D_{\omega^k}(X) + \sum_{z \in \omega^k} \sum_{j=1}^{J^k} w_j(z) \mathcal{N}(X; t_{\omega,j}^k, m_j(z), P_j^k)$$

where

$$w_j(z) = \frac{p_D w_{\omega,j}^k q_j(z)}{\lambda c(z) + p_D \sum_{l=1}^{J^k} w_{\omega,l}^k q_l(z)}$$

$$q_j(z) = \mathcal{N}(z; \bar{z}_j, S_j)$$

$$m_j(z) = m_j^k + P_j^{k-1} H_j S_j^{-1} (z - \bar{z}_j)$$

$$P_j^k = P_j^{k-1} - P_j^{k-1} H_j S_j^{-1} H_j^T P_j^{k-1}.$$
\[ D_{\pi_s} (X) = \langle \Psi_1 [w_{o,k}^k, z^k], \rho_{o,k} \rangle (1 - p_D) D_{\omega_a} (X) \]

\[ + \sum_{z \in z^k} \sum_{j=1}^{p_D} w_j (z) \mathcal{N} (X; t_{o,j}^k, m_j^k (z), P_j) \]

where

\[ \Psi_u [w_{o,k}^k, z^k] (n) = \sum_{j=0}^{\min\{n|z|, n-u\}} \left( \begin{array}{c} n \setminus j \setminus u \\ j \end{array} \right) \rho_c (|z| - j) \times (1 - p_D)^{n-j+u} (n-j+u)!
\]

\[ \times e_j \left( \Lambda (w_{o,k}^k, z^k) \right) \]

\[ \Lambda (w_{o,k}^k, z^k) = \left\{ \frac{p_D}{\hat{c} (z)} (w_{o,k}^k)^T q (z) : z \in z^k \right\} \]

\[ w_{o,k}^k = [w_{o,k,1}^k, ..., w_{o,k,L}^k] \]

\[ q (z) = [q_{1} (z), ..., q_{L} (z)] \]

\[ w_j (z) = \frac{p_D w_{o,k}^k q_j (z)}{\hat{c} (z)} \langle \Psi_1 [w_{o,k}^k, z^k \setminus \{z\}], \rho_{o,k} \rangle \]

and \( q_j (z), m_j^k (z) \) and \( P_{k,j} \) are given in Proposition 11.

Proposition 13 is a consequence of Theorem 9 and the Kalman filter update [35]. The GMTCPHD filter update is analogous to the GMCPHD filter [5], with the difference that previous states of the target trajectories are also included.

C. L-scan GMTPHD

In this section, we propose the use of pruning and absorption to limit the number of components in the Gaussian mixture and a computationally efficient implementation of the Gaussian mixture filters: the L-scan GMTPHD and GMTCPHD filters.

The PHD of the GMTPHD/GMTCPHD filters has an increasing number of components as the target state progresses and, in order to limit complexity, we need to bound the number of components. We use the following techniques: pruning with threshold \( \Gamma_p \), setting a maximum number \( J_{max} \) of components and absorption [28]. Absorption consists of removing components that are close to another component with higher weight, and adding the weights of the removed components to the weight of the component that has not been removed. Absorption is motivated by the fact that if two components have a very similar current state, based on a Mahalanobis distance criterion, future measurements will affect both component weights and future states in a similar way. Therefore, without absorption, we would have two components with practically the same Gaussian densities for the trajectory states corresponding to recent time steps, for which we would be repeating the same calculations. It should be noted that absorption is an adaptation of the operation called merging in multitarget filtering [36] to deal with single trajectory densities. The steps of the pruning and absorption algorithms for the GMTPHD/GMTCPHD filters are given in Algorithm 2 where we use the notation \( \Phi_j^k = (w_j^k, t_j^k, m_j^k, P_j^k) \).

In addition, as time progresses, the lengths of the trajectories increase so, eventually, it is not computationally feasible to implement the proposed filters directly. In order to address this problem, we propose the L-scan implementations that propagate the joint density of the states of the last \( L \) time steps and independent densities for the previous states for each component of the PHD. This approach has a Kullback-Leibler divergence interpretation [28] and is motivated by the fact that measurements at the current time step only have a significant impact on the trajectory state estimates for recent time steps.

The L-scan GMTPHD/GMTCPHD filters are implemented as the GMTPHD/GMTCPHD with a minor modification in the prediction step, where we discard the correlations of states that happened \( L \) time steps before the current time step. Given the predicted PHD \( D_{\omega,k} (\cdot) \) in Gaussian mixture form, see (29), its L-scan version is given by approximating the covariance matrices \( P_{\omega,j}^k \) as

\[ P_{\omega,j}^k \approx \text{diag} \left( \tilde{P}_{j,1}^k, \tilde{P}_{j,2}^k, ..., \tilde{P}_{j,L}^k, \tilde{P}_{j,L+1+k}^k \right) \]

where matrix \( \tilde{P}_{j,L+1+k}^k \in \mathbb{R}^{L_n \times L_n \times L_{n_z}} \) represents the joint covariance of the \( L \) last time instants, obtained from \( \tilde{P}_{\omega,j}^k \), and \( \tilde{P}_{j}^k \in \mathbb{R}^{n_z \times n_z} \) represents the covariance matrix of the target state at time \( k \), obtained from \( P_{\omega,j}^k \). Therefore, we have independent Gaussian densities to represent the states outside the L-scan window, and a joint Gaussian density for the states in the L-scan window, as in [37]. The steps of the L-scan GMTPHD and GMTCPHD filters are summarised in Algorithm 2.

It should be noted that the cardinality distribution is not affected by the choice of \( L \) in both filters.

D. Estimation

In this section, we adapt two commonly used estimators of the GMPHD and GMCPHD filters to the GMTPHD and GMTCPHD filters.

1) GMTPHD: We adapt the estimator for the GMPHD filter described in [1] Sec. 9.5.4.4 for sets of trajectories.
Then, the estimated set of trajectories are given by
\[
\hat{\mathbb{X}}_k = \{ (t_{l_1}^k, m_{l_1}^k), \ldots, (t_{l_{N_k}}^k, m_{l_{N_k}}^k) \}
\]
where \( \{ l_1, \ldots, l_{N_k} \} \) are the indices of the PHD components with highest weights.

2) GMTCPHD: We adapt the estimator for the GMCPHD filter described in [1] Sec. 9.5.5.4 for sets of trajectories. The estimated cardinality at time step \( k \) is obtained as
\[
\hat{N}_k = \text{arg max}_{n \in \mathbb{N} | n \geq 0} \rho_{\text{p}+}(n).
\]
Then, the estimated set of trajectories are given by
\[
\hat{\mathbb{X}}_k = \{ (t_{l_1}^k, m_{l_1}^k), \ldots, (t_{l_{N_k}}^k, m_{l_{N_k}}^k) \}
\]
where \( \{ l_1, \ldots, l_{N_k} \} \) are the indices of the PHD components with highest weights.

E. Discussion

In this section, we discuss some of the prominent aspects of the proposed filters. The TPHD/TCPHD filters have a similar structure as the PHD/CPHD filters, with the additional benefit of providing trajectory estimates for the alive targets. That is, at each time step, these filters can estimate the trajectories of each of the targets, including its time of birth. The main difference between the trajectory PHD/CPHD filters and their target counterparts is that the trajectory filters do not integrate out previous states of the trajectories.

The \( L \)-scan GMTPHD/GMTCPHD filters are computationally efficient implementations, which allow the propagation of the posterior for long time sequences. In fact, the 1-scan version \((L=1)\) of the GMTPHD/GMTCPHD filters perform the same computations as the GMPHD/GMCPHD, the only difference being that the trajectory versions store the mean and covariance of the trajectory state at each time instant for each component of the PHD.

It should be noted that the computations of the cardinality probability mass functions and elementary symmetric functions are the same as in the GMCPHD filter so we refer the reader to [5] Sec. V.D for more details. We would also like to note that we have presented the filters considering linear and Gaussian models for ease of exposition. Nevertheless, we can still apply the GMTPHD and GMTCPHD filters for nonlinear measurement and dynamic models by first linearising the system and then applying the prediction and update steps with the linearised model. This is the usual procedure in nonlinear Gaussian filtering, for example, as in the extended Kalman filter, the unscented Kalman filter, or the iterated posterior linearisation filter [35, 38].

VII. Simulations

We proceed to assess the performance of the two proposed filters in comparison with the previous track building procedures for PHD/CPHD filters based on tagging each PHD component [22]. In this track building approach, we estimate the number \( n_k \) of targets as indicated in Section VI-D and take the \( n_k \) highest components of the PHD with different tags. These estimates are then appended to the estimated trajectories with the same tag at the previous time step. We refer to these algorithms as tagged PHD/CPHD filters. All units of the quantities in this section are given in the international system.

We consider a target state \( x = [\hat{p}_x, \hat{\dot{p}}_x, \hat{p}_y, \hat{\dot{p}}_y]^T \), which contains position and velocity. The parameters of the single-target dynamic process are
\[
F = I_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}, \quad Q = q I_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix}
\]
where \( \tau = 0.5 \) is the sampling time and \( q = 3.24 \) is a parameter. We also set \( p_S = 0.99 \). The parameters of the measurement model are
\[
H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad R = \sigma^2 I_2,
\]
where \( \sigma^2 = 4 \), and \( p_D = 0.9 \). The clutter intensity is \( D_c(z) = \lambda_c \cdot u_A(z) \) where \( u_A(z) \) is a uniform density in region \( A = [0,2000] \times [0,2000] \) and \( \lambda_c = 50 \) is the average number of clutter measurements per scan. The birth process is Poisson with a PHD that is represented by a Gaussian mixture with parameters: \( J^* = 3, \lambda_{\beta,j} = 0.1, P_{\beta,j} = \text{diag}([225,100,225,100]) \) for \( j \in \{1,2,3\} \), \( m_{\beta,1}^k = [85,0,140,0]^T, m_{\beta,2}^k = [-5,0,220,0]^T \) and \( m_{\beta,3}^k = [7,0,50,0]^T \).

We have implemented the \( L \)-scan GMTPHD and GMTCPHD filters with \( L \in \{1,2,5\} \) in a scenario with \( N_s = 100 \) time steps. We use a pruning threshold \( \Gamma_p = 10^{-4} \), absorption threshold \( \Gamma_a = 4 \) and limit the number of components to 30. An exemplar output of the 1-scan TPHD filter and the considered ground truth are shown in Figure 3. At each time step, TPHD and TCPHD filters provide an estimate of the set of present trajectories at the current time. The start and end times of an estimated trajectory do not depend on the choice of \( L \) so the output for any other \( L \) has the same start time and duration, but with a different error.

In the following, we evaluate the performance of the filters by Monte Carlo simulation with \( N_{mc} = 500 \) runs. At each time step \( k \), we measure the error between the set \( \hat{\mathbb{X}}_k \) of alive trajectories and its estimate \( \hat{\mathbb{X}}_k \). In order to do so, we use the metric for sets of trajectories based on linear programming.
However, a more accurate cardinality estimate does not imply just after a target disappears, as can be seen in Figure 5. The TCPHD filter in fact is interesting to note that, in this example, the performance improves estimation performance and lowers the error. It is much higher than for the TPHD filter at most time steps but is remarkably high when targets disappear, which occurs at time steps 70, 80 and 95, see caption of Figure 3. The misdetected target drops, the maximum of the cardinality relation to the TPHD filter is due to the spooky effect at a distance \( d \). The TCPHD filter is able to approximate the cardinality distribution more accurately than the TPHD filter. However, if one target is not detected, the PHD in the area of the misdetected target drops, the maximum of the cardinality distribution of the TCPHD filter may remain the same, and the highest peaks of its PHD can usually be found next to the detected targets. This effect is illustrated in Figure 7. In this situation, the TCPHD filter detects three trajectories and misses the other trajectory. On the other hand, the TCPHD reports four trajectories, but two of them in an area where only one true trajectory lies due to the location of the peaks of the PHD. Therefore, the TCPHD misses the same trajectory as the TPHD, but it also reports an additional false/duplicated trajectory. Nevertheless, in this scenario, the TCPHD performs better in terms of lower cost for missed targets.

The running times of our Matlab implementations of the filters in a computer with a 3.5 GHz Intel Xeon E5 processor are shown in Table I. The running times of the TCPHD filter are roughly 1 second longer than the TPHD for all values of \( L \). Tagged filters have a higher computational burden than TPHD and TCPHD with low \( L \), due to the estimation process that does not work well if two targets are born from the same PHD component at the same time. In order to analyse the results more thoroughly, we make use of the metric decomposition indicated in [37]. We compute the RMS costs in (37) at each time step normalised by \( k \), as in (38). The results are shown in Figure 6. We can first see that changing the value of \( L \) does not change the costs for missed and false targets for the trajectory algorithms, it mainly improves the localisation costs. In fact, the choice of \( L \) affects the localisation errors more than the choice of the TPHD and TCPHD filters, as it has a direct effect on the estimation of past states of the trajectories. In addition, we see that the TCPHD has a lower cost for missed targets, but at the expense of a higher cost for false targets. This cost is much higher than for the TPHD filter at most time steps but is remarkably high when targets disappear, which occurs at time steps 70, 80 and 95, see caption of Figure 3. The explanation of the lower performance of the TCPHD filter in comparison to the TPHD filter is due to the spooky effect at a distance \( d \). The TCPHD filter is able to approximate the cardinality distribution more accurately than the TPHD filter. However, if one target is not detected, the PHD in the area of the misdetected target drops, the maximum of the cardinality distribution of the TCPHD filter may remain the same, and the highest peaks of its PHD can usually be found next to the detected targets. This effect is illustrated in Figure 7. In this situation, the TCPHD filter detects three trajectories and misses the other trajectory. On the other hand, the TCPHD reports four trajectories, but two of them in an area where only one true trajectory lies due to the location of the peaks of the PHD. Therefore, the TCPHD misses the same trajectory as the TPHD, but it also reports an additional false/duplicated trajectory. Nevertheless, in this scenario, the TCPHD performs better in terms of lower cost for missed targets.

The RMS trajectory errors for the algorithms are plotted in Figure 4. As expected, for the TPHD and TCPHD, increasing \( L \) improves estimation performance and lowers the error. It is interesting to note that, in this example, the performance of the TCPHD filter is slightly lower than the TPHD filter, even though it is a more accurate algorithm in terms of KLD minimisation in the update step. The TCPHD filter in fact provides a more accurate estimation of the cardinality, except just after a target disappears, as can be seen in Figure 5. However, a more accurate cardinality estimate does not imply that a multitrack/multitrack estimate is more accurate, according to traditional multitrack tracking evaluation methods, in which it is more important to have a low number of false and missed targets [41]. Example 2]. The tagged PHD and CPHD filters provide a considerably higher error, as track estimation is done in a manner that does not work well if two targets are born from the same PHD component at the same time.

\[
d^2(\hat{X}_a, \hat{X}_a) = \sum_{i=1}^{N_d} \alpha_{a,i} \left[ \sum_{k=1}^{\gamma} (\hat{X}_{a,k} \cdot \hat{X}_{a,k}) + \beta_{a,k} \right] + \left[ \sum_{k=1}^{\gamma} (\hat{X}_{a,k} \cdot \hat{X}_{a,k}) + \beta_{a,k} \right].
\]

Figure 3: Exemplar outputs of the TPHD filter at time steps 50 (left) and 70 (right). The dashed blue lines represent the true trajectories. The blue squares denote their starting positions and the numbers next to them starting times. There are four trajectories, two start at time step 1, close to each other, and end at time step 80. The other two start at time steps 5 and 10 and end at time steps 70 and 95, respectively. Black circles represent the current measurements. The TPHD filter is able to estimate the alive trajectories at each time step.

| \( L \) | 1 | 2 | 5 | 10 | 20 | 30 |
|-------|---|---|---|----|----|----|
| TPHD  | 1.1 | 1.1 | 1.2 | 1.7 | 3.4 | 6.0 |
| TCPHD | 2.0 | 2.0 | 2.1 | 2.6 | 4.3 | 6.9 |
| Tagged PHD | 2.2 |
| Tagged CPHD | 3.2 |
Figure 4: RMS error of the alive trajectories for the TPHD/TCPHD filter and tagged PHD/CPHD filters. Filters based on sets of trajectories have a much higher performance than tagged filters. Increasing $L$ lowers the error.

Figure 5: Estimated cardinality against time for the filters. CPHD-based filters estimate the cardinality more accurately except just after a target disappears.

$p$ and $c$, between target states and their estimates at each time step. The trajectory metric and GOSPA return the same errors for TPHD/TCPhD filters, for all simulation parameters considered, which implies that the cost of track switches is negligible. Tagged filters show track switches so the trajectory metric and GOSPA do not coincide. GOSPA errors are not shown due to space constraints, but they are generally 0.02 lower than the trajectory metric error for the tagged filters. The errors of the tagged filters are, in general, considerably higher than for the set of trajectories filters. According to the trajectory metric, TPHD beats TCPHD in four out of the six scenarios considered. This is due to the effect of the duplicated trajectories in CPHD filtering, explained in Figure 7. On the contrary, according to OSPA, the TCPHD is a better alternative than the TPHD for all simulation parameters. The reason for this is that OSPA does not penalise the false trajectory estimates of the TCPHD filter if the estimated cardinality and true cardinality are alike, as illustrated in Figure 7. In any case, we can see that for all metrics, the two algorithms proposed in this paper, TPHD and TCPHD outperform PHD/CPHD mechanisms for track building based on tags.

Figure 6: RMS costs for localisation errors for properly detected targets, missed targets, false targets and track switches for the alive trajectories at every time step. The scale of the $y$ axis changes in the figure for the switching cost. The RMS costs for missed targets and false targets do not change with the considered value of $L$. Increasing $L$, decreases the localisation cost for trajectory-based algorithms. TCPHD misses fewer targets than TPHD, but with an increase in false targets. Switching cost is negligible for TPHD/TCPHD filters. The tagged PHD and CPHD filters have a high cost for missed targets and show track switching.
Table II: Error in alive trajectories averaged over all time steps for the trajectory metric (TM) and OSPA

| L | TM/GOSPA | OSPA | TM/GOSPA | OSPA | Tagged PHD | Tagged TCPHD |
|---|---|---|---|---|---|---|
| No change | 5.71 | 5.08 | 4.88 | 4.03 | 3.74 | 3.65 | 6.11 | 5.50 | 5.31 | 3.93 | 3.63 | 3.53 |
| σ^2 = 16 | 9.40 | 8.60 | 7.15 | 5.87 | 5.45 | 5.22 | 9.65 | 8.83 | 8.37 | 5.81 | 5.36 | 5.12 |
| σ^2 = 1 | 4.69 | 4.40 | 3.36 | 3.53 | 3.40 | 3.39 | 5.13 | 4.86 | 4.81 | 3.42 | 3.28 | 3.27 |
| λx = 70 | 5.80 | 5.19 | 3.00 | 4.10 | 3.82 | 3.73 | 6.15 | 5.56 | 5.37 | 4.01 | 3.71 | 3.60 |
| p_d = 0.99 | 4.16 | 3.36 | 3.17 | 2.64 | 2.22 | 2.10 | 4.10 | 3.28 | 3.02 | 2.59 | 2.16 | 2.04 |
| p_d = 0.85 | 7.47 | 6.96 | 6.79 | 4.95 | 4.71 | 4.64 | 6.93 | 6.36 | 6.16 | 4.49 | 4.21 | 4.11 |

Figure 7: Estimated trajectories (in red) for the TPHD filter (left) and TCPHD filter (right) at time step 44 and L = 1 in an illustrative run. The ground truth for the four alive trajectories is shown as dashed blue lines. Current measurements are denoted as black circles. One trajectory does not generate a measurement at the current time step and the TPHD and TCPHD filter do not detect it. The TPHD and TCPHD filters estimate the other three trajectories well. However, the TCPHD filter also reports an extra, false trajectory that coincides with the estimated trajectory that starts around location (0, 230) at all time steps, except at the current time step in which it is slightly different. The sum of GOSPA/OSPA cost at all time steps (normalised by \( \sqrt{T} \)) are for TPHD 8.16/5.76 and for TCPHD 10.35/5.76. According to OSPA, TPHD and TCPHD estimates are of the same accuracy despite the additional false trajectory in the TCPHD output. GOSPA error which coincides with the trajectory metric error in this case, ranks the algorithms in terms of localisation errors, and costs for missed and false targets.

VIII. CONCLUSIONS

In this paper we have developed the trajectory PHD and CPHD filters based on KLD minimisations and sets of trajectories. The TPHD and TCPHD filters propagate a Poisson multitrajectory density and an IID cluster multitrajectory density through the filtering recursion to make inference on the set of alive trajectories. The theory found in this paper endows the PHD/CPHD filters with the capability of estimating trajectories from first principles, which can span their already widespread use to more applications where tracks are required.

We have also proposed a Gaussian mixture implementation of the filters. In particular, the parameter L of the L-scan versions governs the accuracy of the estimation of past states of the trajectories. We have analysed the proposed filters in terms of localisation errors, missed targets and false targets, and also based on OSPA. Increasing L mainly improves the localisation error of past states of the trajectory for both filters. In general, the TCPHD filter misses fewer targets but reports more false targets than the TPHD filter, so the relative performance of both filters with the considered estimators depends on the considered simulation.

REFERENCES

[1] R. P. S. Mahler, Advances in Statistical Multisource-Multitarget Information Fusion. Artech House, 2014.
Supplementary material of “Trajectory PHD and CPHD filters”

APPENDIX A

In this appendix we prove Theorem 4. A multitrajectory density \( \pi(\cdot) \) can be written as [31]
\[
\pi\left(\{X_1, \ldots, X_n\}\right) = \rho_\pi(n) \frac{n!}{\pi_n(X_1, \ldots, X_n)}
\] (39)
where \( \pi_n(\cdot) \) is a permutation invariant ordered trajectory density such that
\[
\int \pi_n(X_1, \ldots, X_n) \, dX_{1:n} = 1.
\]
The marginal density of this density is written as
\[
\tilde{\pi}_n(X) = \int \pi_n(X, X_2, \ldots, X_n) \, dX_{2:n} = \frac{1}{\rho_\pi(n)} \int \pi(\{X, X_2, \ldots, X_n\}) \, dX_{2:n}.
\]
Substituting (11) and (39) into (15), we get
\[
\Xi \pi_n(X_1, \ldots, X_n) \, dX_{1:n} = 1.
\]
where
\[
\pi_n(X_1, \ldots, X_n) = \frac{1}{\rho_\pi(n)} \int \pi(\{X, X_2, \ldots, X_n\}) \, dX_{2:n}.
\]
Substituting (11) and (39) into (15), we get
\[
D(\pi \mid \nu) = \sum_{n=0}^{\infty} \rho_\pi(n) \log \frac{\rho_\pi(n)}{\rho_\nu(n)} + \sum_{n=0}^{\infty} \rho_\pi(n) \int \pi_n(X_1, \ldots, X_n) \times \log \frac{\pi_n(X_1, \ldots, X_n)}{\prod_{j=1}^{n} \tilde{\nu}(X_j)} \, dX_{1:n}.
\] (40)
The objective is to find \( \rho_\nu(\cdot) \) and \( \tilde{\nu}(\cdot) \) that minimise \( D(\pi \mid \nu) \). From KLD minimisation over discrete variables, it is clear that \( \rho_\nu(\cdot) = \rho_\pi(\cdot) \) minimises the KLD. Minimizing \( D(\pi \mid \nu) \) w.r.t. \( \tilde{\nu}(\cdot) \) is equivalent to minimising the functional
\[
L[\tilde{\nu}] = -\int \sum_{n=0}^{\infty} \rho_\pi(n) \, n \tilde{\pi}_n(X) \log \tilde{\nu}(X) \, dX.
\]
which is minimised by [28]
\[
\tilde{\nu}(X) = \frac{D_\pi(X)}{\sum_{n=0}^{\infty} \rho_\pi(n) \, n},
\]
or equivalently, \( D_\nu(\cdot) = D_\pi(\cdot) \).

APPENDIX B

In this appendix, we prove [16]. The IID cluster multitrajectory density \( \nu^k(\cdot) \) which minimises the KLD has the same cardinality distribution as \( \pi^k(\cdot) \). The cardinality distribution of \( \pi^k(\cdot) \) is such that \( \rho_\pi^k(m) = 1 \), so we compute its PHD. The posterior can be written as [43, Eq. (12.91)]
\[
\pi^k(\{X_1, \ldots, X_m\}) = \sum_{\sigma \in \Xi_m} \prod_{i=1}^{m} \tilde{d}_i(X_{\sigma_i}) \prod_{i=1}^{m} \tilde{d}_i(X_{\sigma_i+m})
\] (41)
where \( \Xi_m \) is the set that contains all permutations of \( \{1, \ldots, m\} \) and \( \sigma_i \) is the \( i \)th component of \( \sigma \).

Substituting (41) into (4), the PHD of the posterior is
\[
D_\pi^k(X) = \sum_{n=0}^{\infty} \frac{1}{n!} \int \pi^k(\{X, X_1, \ldots, X_n\}) \, dX_{1:n}
\]
\[
= \sum_{i=1}^{m} \tilde{d}_i(X) + \sum_{i=1}^{m} \tilde{d}_i(X),
\]
which is equal to \( D_\nu^k(X) \). Then, using (12), we have that the single trajectory density \( \hat{\nu}^k(\cdot) \) is
\[
\hat{\nu}^k(X) = \frac{D_\nu^k(X)}{m} = \frac{\sum_{i=1}^{m} \tilde{d}_i(X)}{m} + \sum_{i=1}^{m} \tilde{d}_i(X).
\]
A single trajectory with this density exists at the current time \( k \) with probability \( m_a/m \). As we have \( m \) IID trajectories with this distribution, the number of alive trajectories follows the binomial distribution in [16].

APPENDIX C

In this appendix, we prove Theorem 9 (TCPHD update). The proof is quite similar to the CPHD update [16].

A. Density of the measurement

As the current set of targets is an IID cluster with cardinality \( \rho_{\omega^k}(\cdot) \) and PHD \( D_{\omega^k}(\cdot) \), the density of the measurements, under Assumptions U1, U4 and U5, is the same as in the CPHD filter [16, Eq. (32)].
\[
\ell^k(z^k) = \langle \rho_{\omega^k}, Y_0 \, D_{\omega^k}, z^k \rangle \prod_{z \in \mathbf{z}^k} \tilde{c}(z).
\] (42)

B. Cardinality of the posterior

As the set of trajectories present at the current time has the same cardinality as the set of targets at the current time, the updated cardinality distribution is the same in both cases so the TCPHD cardinality update is the same as the CPHD filter cardinality update, which is given in [24].

C. PHD of the posterior

The PHD of the posterior is
\[
D_{\pi^k}(X) = \int \pi^k(\{X\} \cup X) \, dX
\]
\[
= \frac{1}{\ell^k(z^k)} \int \ell^k(z^k | \pi^k(\{X\} \cup X)) \omega^k(\{X\} \cup X) \, dX
\]
\[
= \frac{1}{\ell^k(z^k)} \sum_{n=0}^{\infty} (n+1) \int \ell^k(z^k | \pi^k(\{X_1, \ldots, X_n\})) \times \rho_{\omega^k}(n+1) \tilde{\omega}^k(X) \prod_{j=1}^{n} \tilde{\omega}^k(X_j) \, dX_{1:n}.
\]
We apply the decomposition [16, Eq. (14)]
\[
\ell^k(z^k | \pi^k(\{X\}) \cup \tau^k(X)) = (1 - p_D(\tau^k(\{X\})) \ell^k(z^k | \tau^k(X))
\]
+ \rho_{\ell}^k (m) = \sum_{j=0}^{m} \rho_c (m - j) \sum_{n=0}^{\infty} \binom{n}{j} \rho_{\omega_k} (n) \times \left[ \int (1 - P_D (x)) D_{\omega_k} (x) dx \right]^{n-j} \times \left[ \int P_D (x) D_{\omega_k} (x) dx \right]^j.

By changing \rho_c (\cdot), P_D (\cdot) and \omega_k (\cdot) for \rho_{\beta_k} (\cdot), \gamma (\cdot) and \pi_k (\cdot), respectively, we obtain (23).

\section*{B. PHD}

The result for the PHD of the TCPHD prediction, which is the same as for the TPHD prediction, see (19), can be established directly based on thinning, displacement and superposition of point processes, as in the CPHD filter for targets \cite{16}. Nevertheless, for completeness, we provide more details regarding its calculation here.

We have a multitrajectory density \( \pi_k^{-1} (\cdot) \) with a PHD \( D_{\pi_k^{-1}} (\cdot) \) and we want to compute the PHD of the multitrajectory density \( \omega_k (\cdot) \), which is \( D_{\omega_k} (\cdot) \). We first compute the PHD of \( \xi_k (\cdot) \), which represents the multitrajectory density of the surviving trajectories and is given by

\[ \xi_k (X) = \int f_s (X | Y) \pi_k^{-1} (Y) \delta Y \]  

where \( f_s (\cdot | \cdot) \) is the transition multitrajectory density for surviving trajectories. We proceed to explain a decomposition of \( f_s (\cdot | \cdot) \), which will be useful for the proof.

We first write the probability of survival and single trajectory transition density for alive trajectories at time \( k \) as a function of trajectories. These correspond to

\[ p'_s (t, x^{1:i}) = p_s (x^i) \]  

\[ g' (t, x^{1:i} | t', z^{1:i'}) = \delta [t - t'] g (x^i | z^{1:i'}) \delta (x^{1:i-1} - z^{1:i'}) \]

where \( t + i - 1 = k \) and \( t' + i' - 1 = k - 1 \), otherwise, \( p'_s (\cdot) \) and \( g' (\cdot | \cdot) \) are zero. Also, \( \delta (\cdot) \) and \( \delta (\cdot) \) denote Kronecker and Dirac delta, respectively.

We use the following decomposition for the transition density of surviving trajectories

\[ f_s (\{ X \} \cup X | Y) = \sum_{Y \in \mathcal{Y}} g' (X|Y) p'_s (Y) f_s (X|Y \setminus \{ Y \}) . \]

Note that this decomposition results from the fact that given the surviving trajectories \( \{ X \} \cup X \), trajectory X must have survived from a trajectory Y that belongs to Y and the rest of the surviving trajectories X have survived from the rest of the trajectories at the previous time step Y \setminus \{ Y \}.

The PHD of the surviving trajectories is obtained by substituting (45) into (4)

\[ D_{\xi_k} (X) = \int \xi_k (X \cup \{ X \}) \delta X \]
\[
= \int \left[ \int f_s (\{X\} \cup X|Y) \delta X \right] \pi^{k-1} (Y) \delta Y.
\]

We simplify the inner integral in (48) as

\[
\int f_s (\{X\} \cup X|Y) \delta X
= \int \sum_{Y \in Y} g' (X|Y) p'_s (Y) f_s (X|Y \setminus \{Y\}) \delta X
= \sum_{Y \in Y} g' (X|Y) p'_s (Y).
\]

Substituting this equation into (48) and applying Campbell’s theorem [1, Sec. 4.2.12]

\[
D_{\xi_k} (X) = \int \left( \sum_{Y \in Y} g' (X|Y) p'_s (Y) \right) \pi^{k-1} (Y) \delta Y
= \int g' (X|Y) p'_s (Y) D_{\pi^{k-1}} (Y) \, dY.
\]

Substituting (46) and (47) into (49) yields \( D_{\xi_k} (\cdot) \) in (19). By the superposition theorem of point processes [44], the PHD of \( \omega_k (\cdot) \) is the sum of the PHDs of the surviving trajectories, which is given by \( D_{\xi_k} (\cdot) \), and the PHD of the new born trajectories, which is given by \( D_{\beta_k} (\cdot) \). Therefore, the TCPHD prediction equation for the PHD is the same as for the TPHD filter, see (19).