

Article

GLMB Tracker with Partial Smoothing

Tran Thien Dat Nguyen ^{1,*}  and Du Yong Kim ² ¹ School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley 6102, Australia² School of Engineering, RMIT University, Melbourne 3000, Australia

* Correspondence: t.nguyen172@postgrad.curtin.edu.au

Received: 9 August 2019; Accepted: 10 October 2019; Published: 12 October 2019



Abstract: In this paper, we introduce a tracking algorithm based on labeled Random Finite Sets (RFS) and Rauch–Tung–Striebel (RTS) smoother via a Generalized Labeled Multi-Bernoulli (GLMB) multi-scan estimator to track multiple objects in a wide range of tracking scenarios. In the forward filtering stage, we use the GLMB filter to generate a set of labels and the association history between labels and the measurements. In the trajectory-estimating stage, we apply a track management strategy to eliminate tracks with short lifespan compared to a threshold value. Subsequently, we apply the information of trajectories captured from the forward GLMB filtering stage to carry out standard forward filtering and RTS backward smoothing on each estimated trajectory. For the experiment, we implement the tracker with standard GLMB filter, the hybrid track-before-detect (TBD) GLMB filter, and the GLMB filter with objects spawning. The results show improvements in tracking performance for all implemented trackers given negligible extra computational effort compared to standard GLMB filters.

Keywords: labeled RFS; RTS smoother; GLMB filter

1. Introduction

While single-object tracking algorithms have been studied extensively for more than half a century, multi-object tracking is currently a trending topic in signal processing society due to its extensive applications. The challenges of the multi-object tracking problem arise in the context of miss-detection, false alarms, object thinning, and appearing processes. To tackle these problems, several frameworks have been put forward in the literature such as the Joint Probabilistic Data Association (JPDA) [1], multiple hypotheses tracking [2], and recently, Random Finite Sets (RFS) [2]. In particular, RFS forms the mathematical basis of many modern multi-object filters such as Probability Hypothesis Density (PHD) filter [3–7], cardinalized PHD (CPHD) filter [8–10], multi-Bernoulli filter [11,12], the Generalized Labeled Multi-Bernoulli (GLMB) filter [13–19], and its approximation the Labeled Multi-Bernoulli (LMB) filter [20,21]. In many applications, tracking algorithms rely on the standard point measurements to update the object states; in contrast, TBD [22–25] is an alternative approach that bypasses the detection module to directly exploit the observed spatial data. This technique is introduced under the RFS framework in Reference [26] with the development of the so-called separable likelihood model and, recently, in a hybrid (combination of standard observation and separable observation models) approach in Reference [27]. In terms of system modelling, in many multi-object tracking scenarios, it is sufficient to consider object thinning and appearing processes via survivals, deaths, and instantaneous birth models. However, in many practical applications, new objects are also generated from a set or a subset of existing objects. In the context of RFS-based filtering techniques, such spawning models have been proposed for CPHD filter in References [28,29] and for GLMB filter in Reference [30]. Because these spawning models correctly reflect the physical state of the systems with spawning objects, the accuracy of the estimate is improved.

The early works on practical smoothing algorithms for single-object tracking were introduced by Bryson and Frazier [31]; by Rauch, Tung, and Striebel [32]; and subsequently by Fraser and Potter [33]. Later on, many alternative smoothing algorithms for a nonlinear dynamic model were proposed in References [34–39]. Recently, the closed-form solution for the Gaussian Mixture (GM) forward-backward smoother was derived in Reference [3]. Furthermore, the smoother for the multi-sensor tracking problem is addressed in Reference [40], while the smoothing solution for maneuvering-object tracking is presented in Reference [41]. In Reference [42], a method for joint tracking smoothing of object trajectory based on function fitting is also proposed. In a multi-object tracking context, several smoothing techniques have been put forward in the literature despite the challenge of the large smoothing state space. In particular, smoothing for PHD filter is introduced in References [43–45], while smoothing for CPHD filter and Multi-Bernoulli filter are given respectively in References [46,47]. Multiple objects can also be tracked with a fixed-lag smoother via Interacting Multiple Model (IMM) in Reference [48]. Closed-form solution for forward-backward smoothing based on GLMB RFS is introduced in Reference [49], and recently, multi-scan smoothing technique was proposed in Reference [50] with an efficient implementation based on Gibbs sampling, which can easily handle 100 scans. This is an unprecedented advance over traditional multi-scan solutions, which can only handle about 10 scans.

The labeled RFS approach has several theoretical and practical advantages over unlabeled approaches. The first is that labeled RFS filters can provide trajectory estimates naturally without heuristics, whereas this is not possible with unlabeled RFS filters; see Reference [51]. The second is that labeled RFS can provide ancestry information in a principled manner, whereas unlabeled RFS does not have the mechanism to do this (even with smoothing) [30]. The third is that labeled RFS admits analytical solutions such as the GLMB densities that are still valid RFS densities after any truncation, whereas unlabeled RFS cannot; see [51]. The fourth is that the truncation error (or error bound) for labeled RFS, such as GLMB, is available analytically, whereas this is not available for unlabeled RFS [13]. Consequently, truncation-based unlabeled RFS algorithms are heuristics [51]. Numerically, labeled RFS filters such as the GLMB have been demonstrated to be scalable in the number of objects [52], number of scans [50], and number of sensors [53]. Hence, the GLMB is a versatile class of models for multi-sensor multi-object problems.

In this paper, we introduce a tracker based on the GLMB filters and a modification of the multi-scan estimator proposed in Reference [50]. After the forward GLMB filtering stage, a pre-smooth stage is implemented to eliminate short-term tracks, which are usually initiated by false births or spawns. The threshold to prune these tracks varies depending on the tracking scenario. Subsequently, a multi-scan estimator which consists of a standard single-object filter and an Rauch–Tung–Striebel (RTS) smoother is applied on each estimated trajectory to produce smoothed estimates. As the proposed multi-scan estimator operates only on the estimated trajectories, the complexity is much lower than the full smoothing solution proposed in Reference [50]. Especially, this proposed tracker can completely eliminate track fragmentation as the multi-scan estimator estimates the entire trajectories but not single-scan multi-object states as in standard GLMB filters [13,15,16]. We demonstrate the application of the proposed tracker on both a standard measurement model and a TBD measurement model as well as tracking scenario with object spawning.

The structure of this paper is as follows. In Section 2, we provide background information on labeled RFSs, the multi-object transition kernel, the observation models, and the single-object RTS smoother for linear and nonlinear dynamic models. In Section 3, we propose the tracker based on the GLMB filters and the multi-scan estimator. In Section 4, we first show the experimental results for tracking with the standard observation model in linear and nonlinear tracking scenarios. We then show the tracking results of the proposed algorithm with a hybrid observation model. Finally, we demonstrate the performance of the algorithm on tracking biological cells in an image sequence where spawning process occurs.

2. Background

2.1. The Labeled RFS

Throughout this article, we adhere to the following notations. The set exponential is denoted as $[h(\cdot)]^X = \prod_{x \in X} h(x)$ while the inner product notation is denoted as $\langle f, g \rangle = \int f(x)g(x)dx$. The generalization of the Kronecker delta is denoted as follows:

$$\delta_Y(X) = \begin{cases} 1 & X = Y \\ 0 & X \neq Y \end{cases}$$

The set inclusion function is written as follows:

$$1_Y(X) = \begin{cases} 1 & X \subseteq Y \\ 0 & \text{otherwise} \end{cases}$$

\mathbf{X} denotes the labeled set of objects, while $\mathbf{x} = (x, l)$ denotes a single labeled object, specifically, $x \in \mathbb{X}$ and $l \in \mathbb{L}$, where \mathbb{X} and \mathbb{L} are respectively the kinematic state space and the discrete labels space at the current time step. \mathcal{L} is a label extraction function, i.e., $\mathcal{L}(\mathbf{x}) = l$ and $\mathcal{F}(\mathbf{X})$ denote sets of finite subsets of \mathbf{X} . The “+” sign is used to indicate the next time step when applicable.

The Finite-Set Statistics (FISST) integration is defined as follows [54]:

$$f(\mathbf{X})\delta\mathbf{X} = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathbb{X}^i} f(\{x_1, \dots, x_i\})d(x_1, \dots, x_i)$$

In multi-object tracking problem, the cardinality of object sets varies when objects enter or leave the surveillance region. As RFS is a random set of points in the sense that the number of points in the set is random and the points themselves are also random and unordered [54], a set of random objects can be naturally characterized as a RFS. Being introduced systematically for the first time in Reference [13], the labeled RFS incorporates the identities of elements into the unlabeled counterpart. Precisely, with the state space \mathbb{X} and marks space \mathbb{L} , the labeled RFS is a marked simple point process whereas each realization has a distinct label [13,15]. The distinct label property is satisfied when \mathbf{X} has the same cardinality as its labels $\mathcal{L}(\mathbf{X})$. Given this, the distinct label indicator can be written as follows [16]:

$$\Delta(\mathbf{X}) = \delta_{|\mathbb{X}|}(|\mathcal{L}(\mathbf{X})|) \quad (1)$$

The introduction of labeled RFS to the multi-object tracking problem allows direct estimation of trajectories which cannot be done previously with conventional RFS without a separate labeling scheme.

2.2. The Multi-Object Transition Kernel

In standard tracking scenario, an existing object can either survive or die in the next time step. The surviving objects are modeled as an LMB RFS with a survival probability of $p_S(x, l)$, a disappearance probability of $q_S(x, l) = 1 - p_S(x, l)$, and a spatial distribution of $f_{S+}(x_+|x, l)$. The model for such surviving objects is given as follows [13,15,16]:

$$\mathbf{f}_{S+}(\mathbf{X}_{S+}|\mathbf{X}) = \Delta(\mathbf{X})\Delta(\mathbf{X}_{S+})1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{X}_{S+}))[\Phi_{S+}(\mathbf{X}_{S+}|\cdot)]^{\mathbf{X}} \quad (2)$$

where

$$\Phi_{S+}(\mathbf{X}_{S+}|x, l) = \sum_{(x_+, l_+) \in \mathbf{X}_{S+}} \delta_l(l_+)p_S(x, l)f_{S+}(x_+|x, l) + [1 - 1_{\mathcal{L}(\mathbf{X}_{S+})}(l)]q_S(x, l)$$

In addition, the new birth objects can instantaneously appear at each time steps and they are modeled with LMB RFS as follows [13,15,16]:

$$\mathbf{f}_{B+}(\mathbf{X}_{B+}) = \Delta(\mathbf{X}_{B+})w_B(\mathcal{L}(\mathbf{X}_{B+})) [p_{B+}]^{\mathbf{X}_{B+}} \quad (3)$$

$$w_B(\mathcal{L}(\mathbf{X}_{B+})) = 1_{\mathbb{B}_+}(\mathcal{L}(\mathbf{X}_{B+})) [1 - r_{B+}]^{\mathbb{B}_+ - \mathcal{L}(\mathbf{X}_{B+})} [r_{B+}]^{\mathcal{L}(\mathbf{X}_{B+})}$$

Furthermore, in certain scenarios, new objects can also be generated from existing objects, which leads to the need of a spawning model in order to correctly predict the state of the system at the next time step. Recently, a spawning model for GLMB filter has been proposed in Reference [30]; we introduce this model again here as follows for the sake of completeness.

For spawned objects, the naming convention is given as follows: if at time step k the label of an object is l , then the spawned labels from l at the next time step is $l_{spawn} = (l, k + 1, i)$, where i is the index to distinguish between different spawned objects from the same parent. Following this convention, the set of all spawned labels in the next time step is $\mathbb{S}_+ = \mathbb{L} \times \{k + 1\} \times \mathbb{N}$, where \mathbb{N} is the set of positive natural numbers [30].

For each spawned object with the label $l_{spawn} \in \mathbb{S}_+(\mathcal{L}(\mathbf{x}))$, it will either exist with the probability $p_T(\mathbf{x}; l_{spawn})$ and a spatial distribution $f_{T+}(x_+ | \mathbf{x}; l_{spawn})$ or not with the probability $q_T(\mathbf{x}; l_{spawn}) = 1 - p_T(\mathbf{x}; l_{spawn})$.

The density of the set \mathbf{P} of new spawned objects from \mathbf{x} is formulated as follows [30]:

$$\mathbf{f}_{T+}(\mathbf{P} | \mathbf{x}, l_{spawn}) = \Delta(\mathbf{P}) 1_{\mathbb{S}_+(\mathcal{L}(\mathbf{x}))}(\mathcal{L}(\mathbf{P})) [\Phi_{T+}(\mathbf{P} | \mathbf{x}; \cdot)]^{\mathbb{S}_+(\mathcal{L}(\mathbf{x}))} \quad (4)$$

where

$$\Phi_{T+}(\mathbf{P} | \mathbf{x}; l_{spawn}) = \sum_{(x_+, l_+)} \delta_{l_{spawn}}(l_+) p_T(x_+, l_{spawn}) f_{T+}(x_+ | \mathbf{x}, l_{spawn}) + [1 - 1_{\mathcal{L}(\mathbf{P})}(l_{spawn})] q_T(x_+, l_{spawn})$$

Let \mathbf{Q} be a labeled set of objects spawned from \mathbf{X} with $\mathcal{L}(\mathbf{Q}) \subseteq \mathbb{S}_+(\mathcal{L}(\mathbf{X}))$. As all labels sets are disjoint, the FISST convolution theorem [54] can be applied.

$$\mathbf{f}_{T+}(\mathbf{Q} | \mathbf{X}) = \Delta(\mathbf{Q}) 1_{\mathbb{S}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{Q})) [\Phi_{T+}(\mathbf{Q} | \cdot)]^{\mathbf{X}} \quad (5)$$

where

$$\Phi_{T+}(\mathbf{Q} | \mathbf{x}) = [\Phi_{T+}(\mathbf{Q} \cap (\mathbb{X} \times \mathbb{S}_+(\mathcal{L}(\mathbf{x}))) | \mathbf{x}; \cdot)]^{\mathbb{S}_+(\mathcal{L}(\mathbf{x}))}$$

As new birth objects (given in Equation (3)) are independent of the previous time step objects, the overall transition model is given as follows:

$$\mathbf{f}(\mathbf{X}_+ | \mathbf{X}) = \mathbf{f}_{S+}(\mathbf{X}_{S+} | \mathbf{X}) \mathbf{f}_{T+}(\mathbf{Q} | \mathbf{X}) \mathbf{f}_{B+}(\mathbf{X}_{B+}) \quad (6)$$

As the spawned objects depend upon the objects from previous time steps, the prediction step of the filtering stage needs to be done in a joint manner to capture the objects' dependency. As a result, approximation is needed to convert the joint object distribution to a standard GLMB density for each time step in order to keep the algorithm tractable.

In the scenario where the spawning process is not present, the multi-object transition kernel is reduced to the following:

$$\mathbf{f}(\mathbf{X}_+ | \mathbf{X}) = \mathbf{f}_{S+}(\mathbf{X}_{S+} | \mathbf{X}) \mathbf{f}_{B+}(\mathbf{X}_{B+}) \quad (7)$$

2.3. The Multi-Object Observation Models

In the RFS multi-object tracking framework, given a set of measurements $Z = \{z_{1:|Z|}\}$, we have a standard observation model of the following form: [54]

$$g(Z|\mathbf{X}) \propto \sum_{\theta \in \Theta(\mathcal{L}(\mathbf{X}))} \prod_{(x,l) \in \mathbf{X}} \psi_Z^{(\theta(l))}(x,l) \quad (8)$$

where

$$\psi_Z^{(\theta(l))}(x,l) = \delta_0(\theta(l))q_D(x,l) + (1 - \delta_0(\theta(l))) \frac{p_D(x,l)g(z_{\theta(l)}|x,l)}{\kappa(z_{\theta(l)})}$$

$\kappa(\cdot)$ is the clutter intensity, $p_D(\cdot)$ and $q_D(\cdot)$ are respectively the detection and miss-detection probabilities, $g(z|x,l)$ is the likelihood that (x,l) generates measurement z , $\theta : \mathbb{L} \rightarrow \{0 : |Z|\}$ is a positive 1-1 map, and Θ is the entire set of such mappings.

For image observation, with the assumption that object template $T(\cdot)$ is not overlapped, i.e., $T(x_1) \neq T(x_2)$ given $x_1 \neq x_2$, the separable likelihood is given by the following [26]:

$$g(y|\mathbf{X}) = f_B(y) \prod_{\mathbf{x} \in \mathbf{X}} g_y(\mathbf{x}) \quad (9)$$

where y denotes the observed image, f_B denotes the likelihood of the entire set of \mathbf{X} , and $g_y(x)$ denotes the likelihood of a single object in the observed image. The designs of f_B and g_y vary according to the applications, characteristics of observed image, and object appearances.

First introduced in Reference [27], the concept of a hybrid TBD observation model takes advantage of both standard and separable likelihood models. Intuitively, while detected objects can be updated by the associated point measurements, the miss-detected objects can be updated directly from the image observation. This intuition can be described mathematically by defining the following:

$$\sigma_T(T(y)|x,l) \triangleq \frac{g_T(T(y)|x,l)}{g_T(T(y)|\emptyset)} \quad (10)$$

The hybrid likelihood can then be written as follows [27]:

$$g(y|\mathbf{X}) \propto \sum_{\theta \in \Theta(\mathcal{L}(\mathbf{X}))} \prod_{(x,l) \in \mathbf{X}} \varphi_y^{(\theta(l))}(x,l) \quad (11)$$

where

$$\varphi_y^{(\theta(l))}(x,l) = \psi^{(\theta(l))}(x,l|Z) [\sigma_T(T(y)|x,l)]^{\delta_0 \theta(l)}$$

2.4. The Single Object RTS Smoother

Given a set single object observation $\{z_{1:N}\}$, where $N \leq K$ with K is the total number of tracking time steps, the smoothed density of an object state at time $k \leq N$, $p(x_k|z_{1:N})$, is obtained as follows [36]. Initially, let the joint distribution of x_k and x_{k+1} be rewritten as follows:

$$p(x_k, x_{k+1}|z_{1:k}) = p(x_{k+1}|x_k)p(x_k|z_{1:k}) \quad (12)$$

Then, the distribution of x_k given x_{k+1} and $z_{1:k}$ is given as follows:

$$p(x_k|x_{k+1}, z_{1:k}) = \frac{p(x_k, x_{k+1}|z_{1:k})}{p(x_{k+1}|z_{1:k})} \quad (13)$$

where $p(x_{k+1}|z_{1:k}) = \int p(x_{k+1}|x_k)p(x_k|z_{1:k})dx_k$

From the Markov state-space model, we have the following property: $p(x_k|x_{k+1}, z_{1:N}) = p(x_k|x_{k+1}, z_{1:k})$. Hence, we have the following:

$$p(x_k|x_{k+1}, z_{1:N}) = \frac{p(x_k, x_{k+1}|z_{1:k})}{p(x_{k+1}|z_{1:k})} \quad (14)$$

Then, the joint distribution of x_k and x_{k+1} given the measurements set $z_{1:N}$ is given as follows:

$$p(x_k, x_{k+1}|z_{1:N}) = p(x_k|x_{k+1}, z_{1:N})p(x_{k+1}|z_{1:N}) \quad (15)$$

Finally, the smoothed density of state x_k can then be obtained via the marginalization step as follows:

$$p(x_k|z_{1:N}) = \int p(x_k|x_{k+1}, z_{1:N})p(x_{k+1}|z_{1:N})dx_{k+1} \quad (16)$$

3. The Proposed Tracker

3.1. The Filtering Stage

For this tracker, we assume Gaussian distribution for the dynamic state of each object. At this first stage, the tracker carries out a standard multi-object filtering process to obtain the forward estimated labels and the measurements to label association history. In this subsection, we provide the forward filtering steps for both the GLMB filter (with standard measurements and hybrid measurements observations) and GLMB filter with object spawning.

3.1.1. GLMB Filter without Objects Spawning

The procedure to estimate the state of a set of objects with the standard GLMB filter without including the spawning model in the transition kernel is given as follows.

Given a GLMB prior [16]

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} \omega^{(I, \xi)} \delta_I(\mathcal{L}(\mathbf{X})) [p^{(\xi)}]^{\mathbf{X}} \quad (17)$$

and the standard observation model as in Equation (8), the filtering density in the next time step is given by the following [16]:

$$\pi_{Z_+}(\mathbf{X}) \propto \Delta(\mathbf{X}) \sum_{I, \xi, I_+, \theta_+} \omega^{(I, \xi)} \omega_{Z_+}^{(I, \xi, I_+, \theta_+)} \delta_{I_+}(\mathcal{L}(\mathbf{X})) [p_{Z_+}^{(\xi, \theta_+)}]^{\mathbf{X}} \quad (18)$$

where $I \in \mathcal{F}(\mathbb{L})$, $\xi \in \Xi$, $I_+ \in \mathcal{F}(\mathbb{L}_+)$, $\theta_+ \in \Theta_+$ where ξ is the tracks to measurement association history and Ξ is the entire space of ξ .

$$\omega_{Z_+}^{(I, \xi, I_+, \theta_+)} = \mathbf{1}_{\Theta_+(I_+)}(\theta_+) [1 - \bar{p}_S^{(\xi)}]^{I-I_+} [\bar{p}_S^{(\xi)}]^{I \cap I_+} [1 - r_{B_+}]^{\mathbb{B}_+ - I_+} [r_{B_+}]^{\mathbb{B}_+ \cap I_+} [\bar{\psi}_{Z_+}^{(\xi, \theta_+)}]^{I_+}$$

$$\bar{p}_S^{(\xi)}(l) = \langle p^{(\xi)}(\cdot, l), p_S(\cdot, l) \rangle$$

$$\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(l_+) = \langle \bar{p}_+^{(\xi)}(\cdot, l_+), \psi_{Z_+}^{(\theta_+(l_+))}(\cdot, l_+) \rangle$$

$$p_{Z_+}^{(\xi, \theta_+)}(x_+, l_+) = \frac{\bar{p}_+^{(\xi)}(x_+, l_+) \psi_{Z_+}^{(\theta_+(l_+))}(x_+, l_+)}{\bar{\psi}_{Z_+}^{(\xi, \theta_+)}(l_+)}$$

$$\bar{p}_+^{(\xi)}(x_+, l_+) = \mathbf{1}_{\mathbb{L}}(\{l_+\}) \frac{\langle p_S(\cdot, l_+) f_{S+}(x_+ | \cdot, l_+), p^{(\xi)}(\cdot, l_+) \rangle}{\bar{p}_S^{(\xi)}(l_+)} + \mathbf{1}_{\mathbb{B}_+}(\{l_+\}) p_{B+}(x_+, l_+)$$

In tracking scenarios where raw spatial detection are also available, the hybrid model in Equation (11) can be used to replace the standard observation model with the probability of miss-detection being scaled by the spatial observation likelihood, i.e., given the GLMB prior as in Equation (17). The filtering density is then given as follows [27]:

$$\pi_{y_+}(\mathbf{X}) \propto \Delta(\mathbf{X}) \sum_{I, \xi, l_+, \theta_+} \omega^{(I, \xi)} \omega_{y_+}^{(I, \xi, l_+, \theta_+)} \delta_{l_+}(\mathcal{L}(\mathbf{X})) [p_{y_+}^{(\xi, \theta_+)}]_{\mathbf{X}} \quad (19)$$

where

$$\omega_{y_+}^{(I, \xi, l_+, \theta_+)} = \mathbf{1}_{\Theta_+(l_+)}(\theta_+) [1 - \bar{p}_S^{(\xi)}]^{I-I_+} [\bar{p}_S^{(\xi)}]^{I \cap I_+} [1 - r_{B_+}]^{\mathbb{B}_+ - I_+} [r_{B_+}]^{\mathbb{B}_+ \cap I_+} [\bar{\varphi}_{y_+}^{(\xi, \theta_+)}]^{I_+}$$

$$\bar{\varphi}_{y_+}^{(\xi, \theta_+)}(l_+) = \langle \bar{p}_+^{(\xi)}(\cdot, l_+), \varphi_{y_+}^{(\theta_+(l_+))}(\cdot, l_+) \rangle$$

$$p_{y_+}^{(\xi, \theta_+)}(x_+, l_+) = \frac{\bar{p}_+^{(\xi)}(x_+, l_+) \varphi_{y_+}^{(\theta_+(l_+))}(x_+, l_+)}{\bar{\varphi}_{y_+}^{(\xi, \theta_+)}(l_+)}$$

3.1.2. GLMB Filter with Objects Spawning

For the prior density which is a GLMB density as in Equation (17) and the transition kernel defined in Equation (6), by applying the joint predict–update approach, a proposal density can be written as follows [30]:

$$\tilde{\pi}_+(\mathbf{X}_+ | Z_+) \propto \Delta(\mathbf{X}_+) \sum_{I, \xi, l_+, \theta_+} \omega^{(I, \xi)} \tilde{\omega}_{Z_+}^{(I, \xi, l_+, \theta_+)} \delta_{l_+}(\mathcal{L}(\mathbf{X}_+)) [\tilde{p}_{Z_+}^{(\xi, \theta_+)}]_{\mathbf{X}_+} \quad (20)$$

$$\tilde{\omega}_{Z_+}^{(I, \xi, l_+, \theta_+)} = [r_{B_+}]^{\mathbb{B}_+ \cap I_+} [1 - r_{B_+}]^{\mathbb{B}_+ - I_+} [\bar{p}_S]^{I \cap I_+} [1 - \bar{p}_S]^{I - I_+} [\bar{p}_T]^{\mathbb{S}_+ \cap I_+} [1 - \bar{p}_T]^{\mathbb{S}_+ - I_+},$$

$$\tilde{p}_{Z_+}^{(\xi, \theta_+)}(x_+, l_+) = \frac{\tilde{p}_+^{(\xi)}(x_+, l_+) \psi_{Z_+}^{(\theta_+(l_+))}(x_+, l_+)}{\tilde{\psi}_{Z_+}^{(\xi, \theta_+)}(l_+)},$$

$$\bar{p}_+^{(\xi)}(x_+, l_+) = \mathbf{1}_{\mathbb{B}_+}(\{l_+\}) p_{B_+}(x_+, l_+) + \mathbf{1}_{\mathbb{L}}(\{l_+\}) \bar{p}_S^{(\xi)}(x_+, l_+) + \mathbf{1}_{\mathbb{S}}(\{l_+\}) \bar{p}_T^{(\xi)}(x_+, l_+),$$

$$\bar{p}_S^{(\xi)} = \frac{\langle p_S(\cdot, l_+) f_{S+}(x_+ | \cdot, l_+), p^{(\xi)}(\cdot, l_+) \rangle}{\bar{p}_S^{(\xi)}(l_+)},$$

$$\bar{p}_T^{(\xi)} = \frac{\langle p_T(l_+) f_{T+}(x_+ | \cdot, l_+), p^{(\xi)}(\cdot, l_+) \rangle}{\bar{p}_T^{(\xi)}(l_+)}$$

$$\bar{p}_S^{(\xi)} = \langle p^{(\xi)}(\cdot, l_+), p_S(l_+) \rangle,$$

$$\bar{p}_T^{(\xi)} = \langle p^{(\xi)}(\cdot, l), p_T(l_+) \rangle,$$

$$\tilde{\psi}_{Z_+}^{(\xi, \theta_+)}(l_+) = \langle \bar{p}_+^{(\xi)}(\cdot, l_+), \psi_{Z_+}^{(\theta_+)}(\cdot, l_+) \rangle.$$

From this proposal density, Gibbs' sampler is applied to select high weight hypotheses. These hypotheses are subsequently used to form a standard GLMB density [30]:

$$\hat{\pi}(\mathbf{X}_+ | Z_+) = \Delta(\mathbf{X}_+) \sum_{I, \xi, I_+, \theta_+} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \hat{\omega}_+^{(I, \xi, I_+, \theta_+)}(Z_+) [p_{B_+} \psi_+^{(\theta_+)}(\cdot | Z_+)]^{\mathbf{X}_{B_+}} \frac{[\hat{p}_+^{(I, \xi, I_+, \theta_+)}(\cdot | Z_+)]^{\mathbf{X}_{S_+} \cup \mathbf{X}_{T_+}}}{[\bar{p}_+^{(I, \xi, I_+, \theta_+)}(\cdot | Z_+)]^{I_+}} \quad (21)$$

$$\hat{\omega}_+^{(I, \xi, I_+, \theta_+)}(Z_+) = \frac{\omega_+^{(I, \xi)}(I_+) [\bar{p}_+^{(I, \xi, I_+, \theta_+)}(\cdot | Z_+)]^{I_+}}{\sum_{I, \xi, I_+, \theta_+} \omega_+^{(I, \xi)}(I_+) [\bar{p}_+^{(I, \xi, I_+, \theta_+)}(\cdot | Z_+)]^{I_+}}$$

$$\hat{p}^{(I, \xi, I_+, \theta_+)}(x_+, l_+ | Z_+) \triangleq \mathbf{1}_{I_+}(\{l_+\}) \int p_+^{(I, \xi, \theta_+)}(\{(x_+, l_+), (x_{1,+}, l_{1,+}), \dots, (x_{n,+}, l_{n,+})\} | Z_+) d(x_{1,+}, \dots, x_{n,+})$$

$$\bar{p}^{(I, \xi, I_+, \theta_+)}(x_+, l_+ | Z_+) \triangleq \mathbf{1}_{\mathbb{B}_+}(\{l_+\}) \langle p_{B_+}(\cdot, l_+), \psi_+^{(\theta_+)}(\cdot | Z_+) \rangle + (1 - \mathbf{1}_{\mathbb{B}_+}(\{l_+\})) \langle \hat{p}_{Z_+}^{(I, \xi, I_+, \theta_+)}(x_+, l_+), \mathbf{1} \rangle$$

3.2. GLMB Multi-Scan Estimator

The concept of a multi-scan estimator is introduced in Reference [50]. Given a multi-scan GLMB from time step j to k , the cardinality distribution of the number of trajectories is given as follows:

$$\Pr(|\mathcal{L}(\mathbf{X}_{j:k})| = n) = \sum_{\xi, I_{j:k}} \delta_n[|I_{j:k}|] w_{j:k}^{(\xi)}(I_{j:k}) \quad (22)$$

One possible form of a multi-scan estimator is to determine the component with the highest weight $w_{j:k}^{(\xi)}(I_{j:k})$ given that it has the most probable cardinality by maximizing Equation (22). The expected trajectory estimate can then be computed from $p_{j:k}^{(\xi)}(\cdot, l)$ for each $l \in I_{j:k}$.

In this work, we proposed modifications to the multi-scan estimator in Reference [50], which can eliminate track fragmentation and improve localization performance. The set of all estimated trajectories is updated at each time step via the most significant hypothesis with the most probable cardinality in the GLMB density. At the time step when state estimation is required, the information of estimated trajectories is passed into the estimator. At this stage, trajectories pruning is applied to eliminate short-term tracks. Subsequently, standard filtering and RTS smoothing techniques are applied on each trajectory to produce smoothed state estimates. The significance of this estimator is that it allows the application of smoothing techniques to improve the tracking accuracy while completely eliminates track fragmentation as the entire trajectory is estimated as a whole. In addition, as the complexity of the estimator depends only on the number of estimated tracks, the additional computational effort of the estimator is negligible compared to GLMB filtering. The detailed implementation of the estimator is given as in following subsections.

3.2.1. Estimating the Trajectories

Given the GLMB density at the end of each filtering cycle, the GLMB filter estimate is the result of the maximum posteriori estimate of the cardinality with the means of the object states being conditioned on the estimated cardinality [15]. Given that the possible highest number of tracked objects is N_{max} , the cardinality distribution of the the objects set over a finite set of hypotheses $\{(I, \xi)_h\}_{h=1:H}$ is written as follows:

$$\rho(n)|_{n=0:N_{max}} = \sum_{\mathfrak{S} \in \{(I, \xi)_h\}_{h=1:H}} \omega^{(\mathfrak{S})} \delta_n(|I^{(\mathfrak{S})}|) \quad (23)$$

The estimated cardinality is given as follows:

$$\hat{N} = \operatorname{argmax}(\rho) \quad (24)$$

The estimated hypothesis is as follows:

$$\hat{\mathfrak{S}} = \operatorname{argmax}_{(\mathfrak{S})} \omega^{(\mathfrak{S})} \delta_{\hat{N}}(|I^{(\mathfrak{S})}|) \quad (25)$$

The information from the filtering stage needs to be captured to facilitate the multi-scan estimator. At this stage, we represent a set of estimated trajectories at time k with a set of tuples defined as $\hat{\mathfrak{T}}_k \triangleq \{(\hat{l}_1^k, \hat{b}_{\hat{l}_1^k}, \hat{\xi}_{\hat{l}_1^k}), \dots, (\hat{l}_N^k, \hat{b}_{\hat{l}_N^k}, \hat{\xi}_{\hat{l}_N^k})\}$, where \hat{l}_n^k is the label of estimated trajectory n at time k , $\hat{b}_{\hat{l}_n^k}$ is its corresponding initial birth state (including the time of birth and initial kinematic state), and $\hat{\xi}_{\hat{l}_n^k}$ is the corresponding association history. In addition, we also have set of tuples for all estimated trajectories $\hat{\mathfrak{T}}$ from time step 1 to current time step k . This set of tuples is updated at the end of each filtering time step via updating the association history and initial birth state of existing trajectories and adding new tuples to the set if the trajectories are new. The procedure to update the tuples set is given in Algorithm 1.

Algorithm 1 Updating trajectories tuples

Input: $\hat{\mathfrak{T}}_k = \{(\hat{l}_1^k, \hat{b}_{\hat{l}_1^k}, \hat{\xi}_{\hat{l}_1^k}), \dots, (\hat{l}_N^k, \hat{b}_{\hat{l}_N^k}, \hat{\xi}_{\hat{l}_N^k})\}$, $\hat{\mathfrak{T}} = \{(\hat{l}_1, \hat{b}_{\hat{l}_1}, \hat{\xi}_{\hat{l}_1}), \dots, (\hat{l}_N, \hat{b}_{\hat{l}_N}, \hat{\xi}_{\hat{l}_N})\}$

Output: The updated trajectories tuples set $\hat{\mathfrak{T}}$

```

for  $n = 1$  to  $\hat{N}$ 
  if  $\hat{l}_n^k \in \{\hat{l}_1, \dots, \hat{l}_N\}$ 
    Replace the tuple of label  $\hat{l}_n^k$  in  $\hat{\mathfrak{T}}$  with  $(\hat{l}_n^k, \hat{b}_{\hat{l}_n^k}, \hat{\xi}_{\hat{l}_n^k})$ 
  else
     $\hat{\mathfrak{T}} = \hat{\mathfrak{T}} \cup (\hat{l}_n^k, \hat{b}_{\hat{l}_n^k}, \hat{\xi}_{\hat{l}_n^k})$ 
  end
end

```

3.2.2. Trajectories Pruning

For a set of estimated trajectories tuples from filtering stage $\hat{\mathfrak{T}} = \{(\hat{l}_1, \hat{b}_{\hat{l}_1}, \hat{\xi}_{\hat{l}_1}), \dots, (\hat{l}_N, \hat{b}_{\hat{l}_N}, \hat{\xi}_{\hat{l}_N})\}$ the lifetime of a trajectory with label \hat{l}_n is the length of the corresponding association history, which is given as follows:

$$\tau(l_n) = \operatorname{length}(\hat{\xi}_{\hat{l}_n}) \quad (26)$$

where $\operatorname{length}(\cdot)$ is the function that determines the length of the vector in its argument. If the length of a track is shorter than the threshold value τ_t , i.e., $\tau(l_n) < \tau_t$, this trajectory will be removed from the set of estimated trajectories.

3.2.3. Numerical Implementation of Single-Object Smoother

For completeness, we outline here the detailed numerical implementation of the single-object RTS smoother for both linear and nonlinear dynamic models with Gaussian assumption on the distribution of the states.

Given a linear dynamic model of the form

$$x_+ = Fx + q, z = Hx + r$$

where x is the system state, F is the linear transformation matrix, H is the linear observation matrix, q and r are respectively the process and observation Gaussian noise, and z is the current time step measurement, the backward smoothing step over an interval $N \leq K$ (where K is the total number of tracking time steps) can be implemented with the standard RTS Smoother [32]. The details of the RTS smoother is given in Algorithm 2, where the superscript s denotes the smoothed results.

Algorithm 2 Single-object Rauch–Tung–Striebel (RTS) smoother

Input: The filtered mean and covariance $\{x_k, P_k\}_{k=1:N}$, F , Q

Output: The smoothed mean and covariance $\{x_k^s, P_k^s\}_{k=1:N}$

Initialization: $x_N^s = x_N$ and $P_N^s = P_N$

for $k = N - 1$ down to 1

$$\bar{x}_{k+1} = Fx_k$$

$$\bar{P}_{k+1} = FP_kF^T + Q$$

$$D = P_{k+1}F(\bar{P}_{k+1})^{-1}$$

$$x_k^s = x_k + D(x_{k+1}^s - \bar{x}_{k+1})$$

$$P_k^s = P_k + D(P_{k+1}^s - \bar{P}_{k+1})D^T$$

end

For a nonlinear dynamic model, the RTS smoother can also be applied via the unscented transformation [39]. Given the dynamic model

$$x_+ = f(x, q), y = h(x, r)$$

where f is the nonlinear state transition function and h is the nonlinear observation function, other variables are interpreted the same as in the linear model; the smoothed results can be inferred via the Unscented RTS (URTS) smoother [36]. The smoothing procedure is presented in Algorithm 3, and the readers are referred to Reference [39] for the detailed implementation of the unscented transform. Compared to the Sequential Monte Carlo method, unscented transform is less computationally expensive as the number of sigma points to approximate a Gaussian distribution is much lower than the number of particles to represent the entire density.

3.2.4. Forward Filtering-Backward Smoothing of Trajectories

In this step, by using the measurement association history, the initial birth information (the state and the time at birth) in the estimated trajectories tuples set, and the measurements set, we apply standard single-object filtering and backward RTS smoothing techniques to produce a set of smoothed distributions of the trajectories. In this work, spatial distributions of tracks are assumed to be Gaussian distributed; hence, the estimated spatial distribution of track labeled l at time k is represented by the mean m_k^l and the covariance P_k^l . The details of the procedure to produce the tracks distributions are given in Algorithm 4. The SingleObjectPrediction and SingleObjectUpdate functions are chosen according to the dynamic model, which can be Kalman prediction and Kalman update or their nonlinear variances. The linearity of the system also determines the SingleObjectSmoothing function, which takes the form of either Algorithm 2 or Algorithm 3 to smooth each individual trajectory. The output of the algorithm is the smoothed spatial distributions of all estimated trajectories, which is $\{m_k^{\hat{i}^n}, P_k^{\hat{i}^n}\}_{k_i^{\hat{i}^n}:k_e^{\hat{i}^n}}$. From this set of distributions, the mean values can be extracted to be used as the estimated states of the trajectories.

Algorithm 3 Single-object Unscented RTS (URTS) smoother**Input:** The filtered mean and covariance $\{x_k, P_k\}_{k=1:N}$, $f(x_+|x)$, Q **Output:** The smoothed mean and covariance $\{x_k^s, P_k^s\}_{k=1:N}$ **Initialization:** $x_N^s = x_N$ and $P_N^s = P_N$ **for** $k = N - 1$ **down to** 1 $\{W_{i-1}^{(m)}, W_{i-1}^{(c)}, [\tilde{X}_i^x; \tilde{X}_i^q]\} = \text{UnscentedTransform}(x_k, P_k, Q)$ $\tilde{X}_{i+} = f(\tilde{X}_i^x, \tilde{X}_i^q)$ $\bar{x}_{k+1} = \sum_i W_{i-1}^{(m)} \tilde{X}_{i+}$ $\bar{P}_{k+1} = \sum_i W_{i-1}^{(c)} (\tilde{X}_{i+} - \bar{x}_{k+1})(\tilde{X}_{i+} - \bar{x}_{k+1})^T$ $\bar{C}_{k+1} = \sum_i W_{i-1}^{(c)} (\tilde{X}_i^x - x_k)(\tilde{X}_{i+} - \bar{x}_{k+1})^T$ $D = \bar{C}_{k+1} (\bar{P}_{k+1})^{-1}$ $x_k^s = x_k + D (x_{k+1}^s - \bar{x}_{k+1})$ $P_k^s = P_k + D (P_{k+1}^s - \bar{P}_{k+1}) D^T$ **end**

While the advantages are mentioned previously, this estimator is also subjected to certain drawbacks in challenging tracking scenarios. First, depending on the nature of the problem, the user needs to set an appropriate pruning threshold τ_l to prevent the estimator from deleting correct trajectories, especially when track identity switching is severe. Second, as the estimator relies on the latest hypothesis to produce estimates, the more this hypothesis deviates from the truth, the more inaccurate the entire estimation. In addition, in the case that wrong new tracks keep appearing in the set of trajectory estimates, overestimating of the number of tracks is also possible. However, the benefit from track fragmentation reduction and improvement of tracking accuracy given negligible computational effort is much more than the risk of incorrectly estimating the number of tracks, and the following simulation results are a strong demonstration of the benefits of our proposed tracker.

Algorithm 4 Trajectory forward filtering-backward smoothing**Input:** $\hat{\mathcal{Z}} = \{(\hat{l}_1, \hat{b}_{\hat{l}_1}, \hat{\xi}_{\hat{l}_1}), \dots, (\hat{l}_N, \hat{b}_{\hat{l}_N}, \hat{\xi}_{\hat{l}_N})\}$, $\{Z_1, \dots, Z_K\}$ **Output:** The estimated trajectories $\{\{m_k^{\hat{l}_1}, P_k^{\hat{l}_1}\}_{k_i^{\hat{l}_1}:k_e^{\hat{l}_1}}, \dots, \{m_k^{\hat{l}_N}, P_k^{\hat{l}_N}\}_{k_i^{\hat{l}_N}:k_e^{\hat{l}_N}}\}$ **for** $n = 1$ **to** N Initialize $\{\bar{m}_{k_i^{\hat{l}_n}}^{\hat{l}_n}, \bar{P}_{k_i^{\hat{l}_n}}^{\hat{l}_n}\}$ from the initial birth $\hat{b}_{\hat{l}_n}$ $\{\bar{m}_{k_i^{\hat{l}_n}}^{\hat{l}_n}, \bar{P}_{k_i^{\hat{l}_n}}^{\hat{l}_n}\} = \text{SingleObjectUpdate}(\bar{m}_{k_i^{\hat{l}_n}}^{\hat{l}_n}, \bar{P}_{k_i^{\hat{l}_n}}^{\hat{l}_n}, z_{\xi_{\hat{l}_n}^{\hat{l}_n}}^{\hat{l}_n})$ **for** k from $k_i^{\hat{l}_n} + 1$ to $k_e^{\hat{l}_n}$ $\{\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}\} = \text{SingleObjectPrediction}(\bar{m}_{k-1}^{\hat{l}_n}, \bar{P}_{k-1}^{\hat{l}_n})$ **if** $\xi_{\hat{l}_n}^k = 0$ $\{\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}\} = \{\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}\}$ **else** $\{\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}\} = \text{SingleObjectUpdate}(\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}, z_{\xi_{\hat{l}_n}^k}^{\hat{l}_n})$ **end****end** $\{m_k^{\hat{l}_n}, P_k^{\hat{l}_n}\}_{k_i^{\hat{l}_n}:k_e^{\hat{l}_n}} = \text{SingleObjectSmoothing}(\{\bar{m}_k^{\hat{l}_n}, \bar{P}_k^{\hat{l}_n}\}_{k_i^{\hat{l}_n}:k_e^{\hat{l}_n}})$ **end**

4. Experimental Results

4.1. Simulation Results

4.1.1. Linear Dynamic Model

In this experiment, we use a constant velocity model for the dynamic of the system. The state vector consists of information regarding the planar position and the velocity of the objects, which is $x_k = [p_x, p_y, \dot{p}_x, \dot{p}_y]^T$; while the measurement vector contains the position of the object, which is $z_k = [z_x, z_y]^T$. The transition and observation models are given respectively as follows:

$$f_+(x_+|x) = \mathcal{N}(x_+; Fx, Q)$$

$$h(z|x) = \mathcal{N}(z; Hx, R)$$

where $F = \begin{bmatrix} I_2 & \Delta I_2 \\ 0_2 & I_2 \end{bmatrix}$, $Q = \sigma_v^2 \begin{bmatrix} \frac{\Delta^4}{4} I_2 & \frac{\Delta^3}{2} I_2 \\ \frac{\Delta^3}{2} I_2 & \Delta^2 I_2 \end{bmatrix}$, $H = \begin{bmatrix} I_2 & 0_2 \end{bmatrix}$, $R = \sigma_\epsilon^2 I_2$. Particularly, in this experiment, we set $\sigma_v = 5$ m/s and $\sigma_\epsilon = 15$ m.

The surveillance region is the $[-1000, 1000] \text{ m} \times [-1000, 1000] \text{ m}$ area, the total time step is $K = 100$, and $\Delta = 1$. The ground truth plot for this experiment is given in Figure 1. The surviving probability is set to $p_S = 0.99$, and the detection probability is $p_D = 0.95$. Clutter rate is set to 66 false alarms per scan. The birth probability is set to $r_B = 0.03$. The states of expected births are $m_B^{(1)} = [0.1, 0, 0.1, 0]^T$, $m_B^{(2)} = [400, 0, -600, 0]^T$, $m_B^{(3)} = [-800, 0, -200, 0]^T$, and $m_B^{(4)} = [-200, 0, 800, 0]^T$. The covariance matrix at birth is $P_B = \text{diag}([10, 10, 10, 10])$. The number of hypotheses for GLMB filter is capped at 20,000 components. In this experiment, we smooth the entire tracking interval from $k = 1$ to $k = K$. The threshold for the smoother to prune the track is set to $\tau_t = 3$ time steps.

We conduct the experiment over 100 Monte Carlo runs. The means of the estimated Optimal Subpattern Assignment (OSPA) error [55] and OSPA² error [52,56] are given respectively in Figures 2 and 3. Figure 4 shows the GLMB filter and proposed tracker-estimated cardinality of objects set for each time step along with the true values.

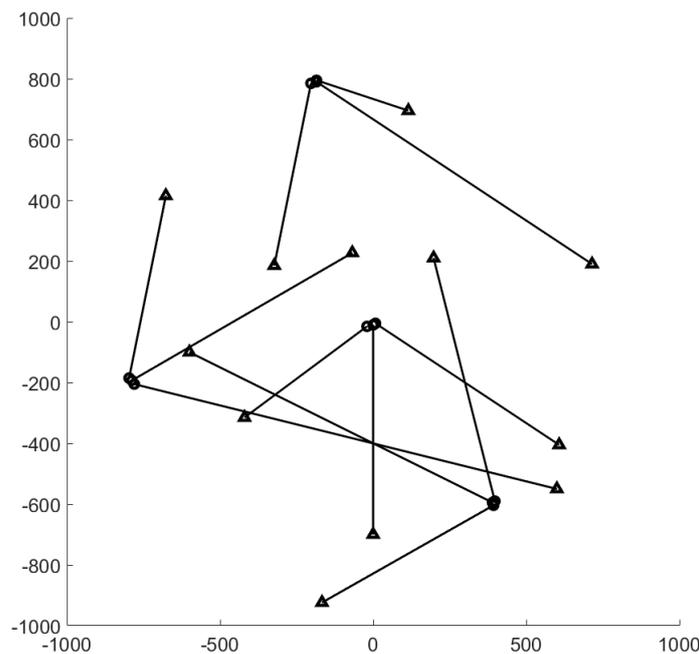


Figure 1. Ground truth for linear dynamic scenario (circle: track start position, triangle: track end position).

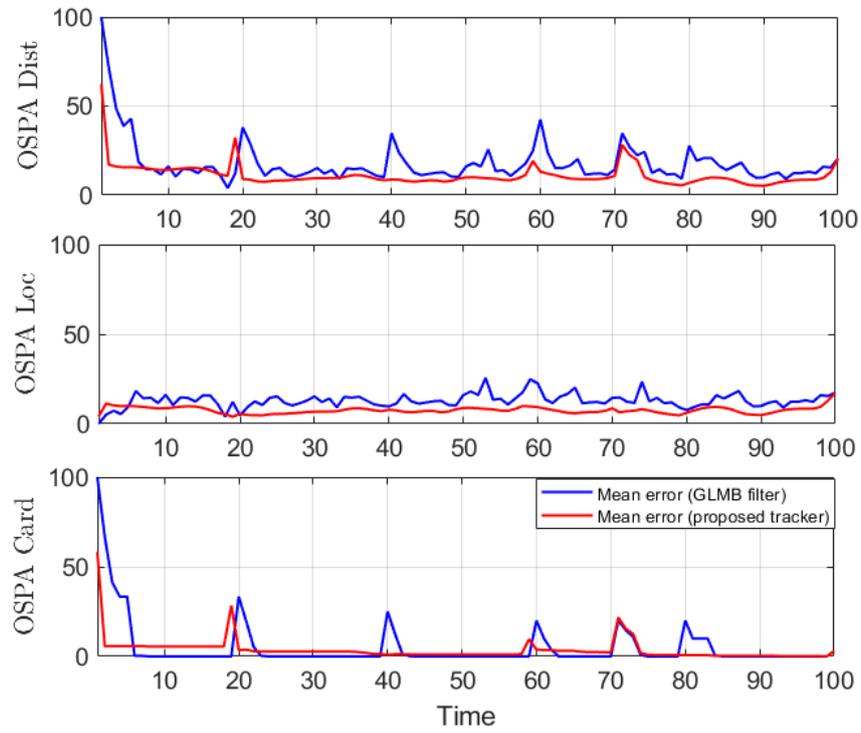


Figure 2. OSPA error for linear dynamic scenario.

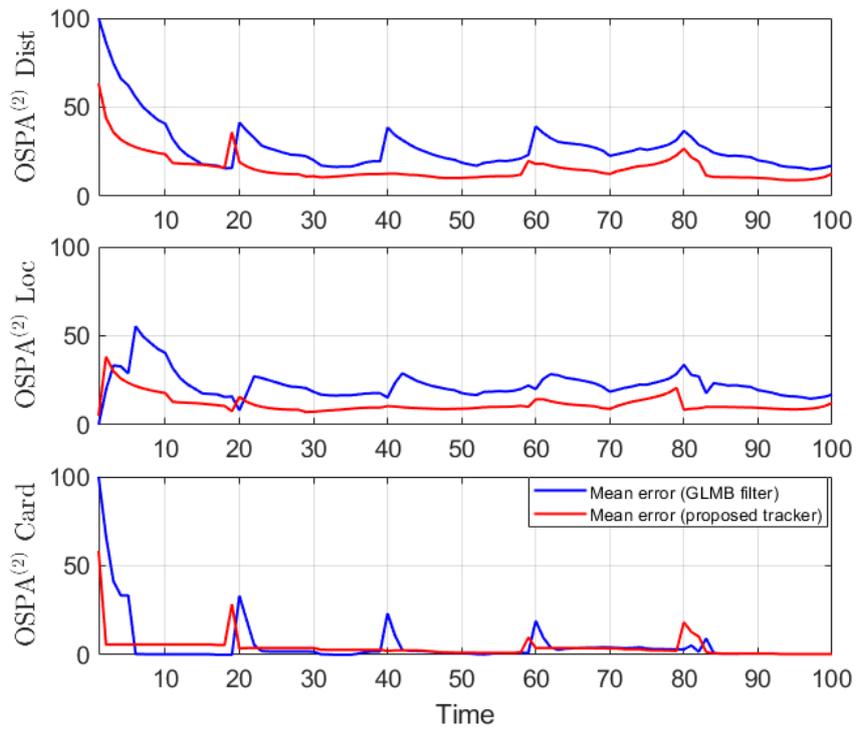


Figure 3. OSPA² error for linear dynamic scenario.

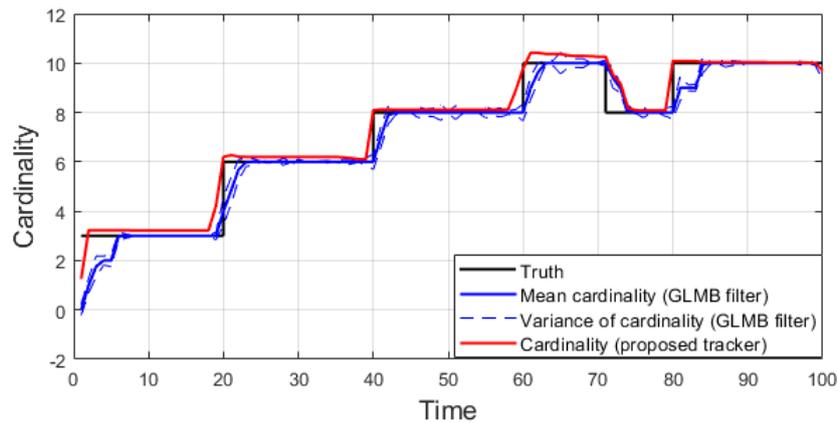


Figure 4. Estimated cardinality for linear dynamic scenario.

4.1.2. Nonlinear Dynamic Model

For the demonstration of the nonlinear tracking scenario, we use a constant turn model with 5-D state vector $x_k = [p_x, p_y, \dot{p}_x, \dot{p}_y, \omega]^T$, where ω is the object's turn rate. The transition density is given as follows:

$$f_+(x_+|x) = \mathcal{N}(x_+; F(\omega)x, Q)$$

$$\text{where } F \left(\begin{bmatrix} p_x \\ \dot{p}_x \\ p_y \\ \dot{p}_y \\ \omega \end{bmatrix} \right) = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta)}{\omega} & 0 & -\frac{1-\cos(\omega\Delta)}{\omega} & 0 \\ 0 & \cos(\omega\Delta) & 0 & -\sin(\omega\Delta) & 0 \\ 0 & \frac{1-\cos(\omega\Delta)}{\omega} & 1 & \frac{\sin(\omega\Delta)}{\omega} & 0 \\ 0 & \sin(\omega\Delta) & 0 & \cos(\omega\Delta) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, Q_\zeta = \begin{bmatrix} \sigma_\omega^2 G G^T & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \text{ and } G = \begin{bmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{bmatrix}.$$

In this experiment, we set $\sigma_\omega = \pi/180$ rad/s and $\sigma_v = 5$ m/s. The observation model is given as the bearing and range detection of the 2D vector $z_k = [\theta, r]^T$ with $\sigma_\theta = \pi/90$ rad and $\sigma_r = 5$ m.

The surveillance region is the half disc of the radius 2000 m with $K = 100$ time steps and $\Delta = 1$. The ground truth for this experiment is given in Figure 5. The surviving probability is set to $p_S = 0.99$ and the detection probability is $p_D = 0.95$. Clutter rate is set to 66 false alarms per scan. The expected birth states are $m_B^{(1)} = [-1500, 0, 250, 0, \pi/180]^T$, $m_B^{(2)} = [-250, 0, 1000, 0, \pi/180]^T$, $m_B^{(3)} = [250, 0, 750, 0, \pi/180]^T$, and $m_B^{(4)} = [1000, 0, 1500, 0, \pi/180]^T$ with $r_B = 0.02$, and the birth covariance is $P_B = \text{diag}([50, 50, 50, 50, \pi/30])$. The number of hypotheses is also capped at 20,000 components. The smoothing interval is the entire tracking sequence from $k = 1$ to $k = K$. We also set the track pruning threshold for the smoother to 3 time steps in this experiment.

For this scenario, we also test the performance of the tracker over 100 Monte Carlo runs. The means of OSPA error and OSPA² error of the estimates are plotted in Figures 6 and 7, respectively, while the set cardinality is shown in Figure 8.

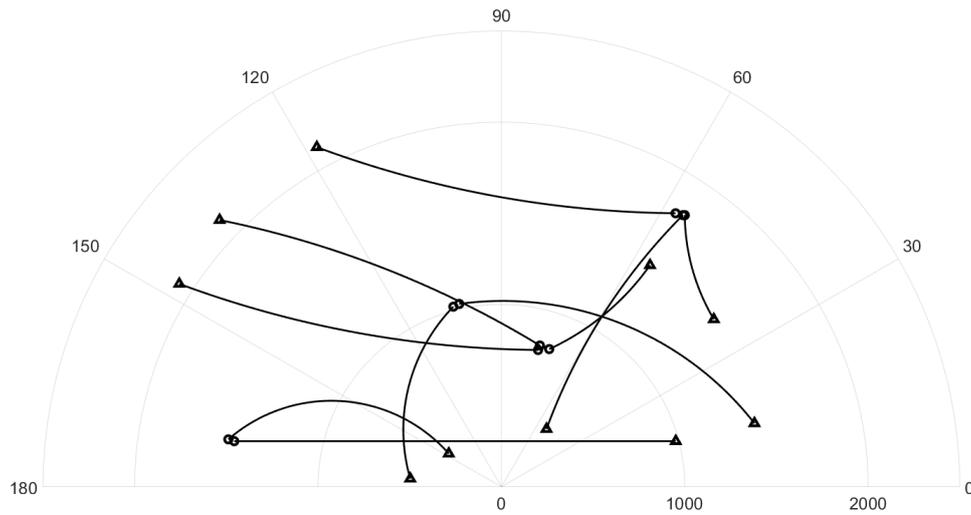


Figure 5. Ground truth for nonlinear dynamic scenario (circle: track start position, triangle: track end position).

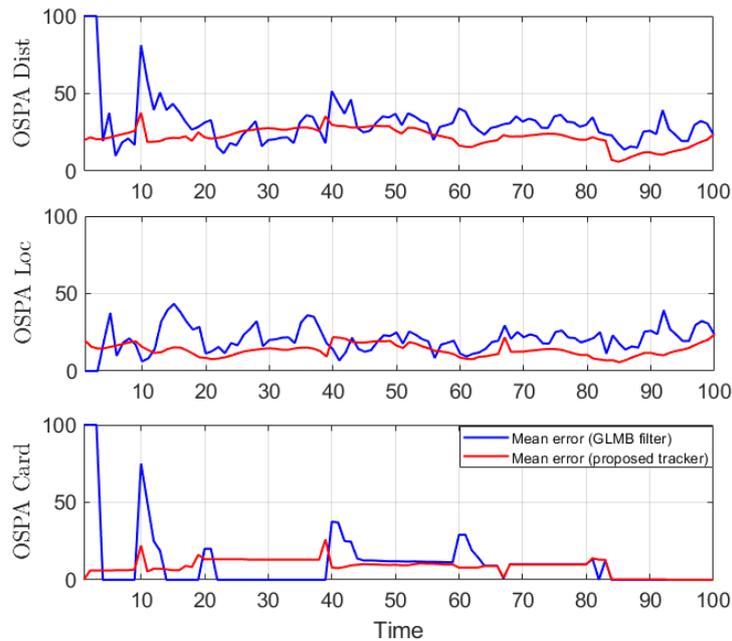


Figure 6. OSPA error for nonlinear dynamic scenario.

4.1.3. Hybrid TBD Observation Model

In this simulation, we use the hybrid measurement model to track objects following a linear dynamic motion model. The surveillance region is 100×100 pixels with image cell size of 1, total time step of $K = 100$, and $\Delta = 1$. The observation are the raw images, which are arrays of pixels. In particular, for a pixel i at the image coordinate $(a^{(i)}, b^{(i)})$, the array value is given as follows [26,27]:

$$y^{(i)} = \left[\sum_{x \in X} \frac{I_k}{2\pi\sigma_h} \exp \left(-\frac{(a^{(i)} - p_x)^2 + (b^{(i)} - p_y)^2}{2\sigma_h^2} \right) \right] + w^{(i)} \quad (27)$$

where $w^{(i)} \sim \mathcal{N}(0, \sigma_y)$ is Gaussian noise. In this experiment, we set $\sigma_h = 4$ and $\sigma_y = 1$. We choose the value of I_k such that the signal to noise ratio (SNR) varies over the range 7 to 10 dB. For the observation model from the perspective of the filter, we fix its SNR value to 10 dB. From the raw images, we then use hard-shareholding to extract the points measurements at each frame.

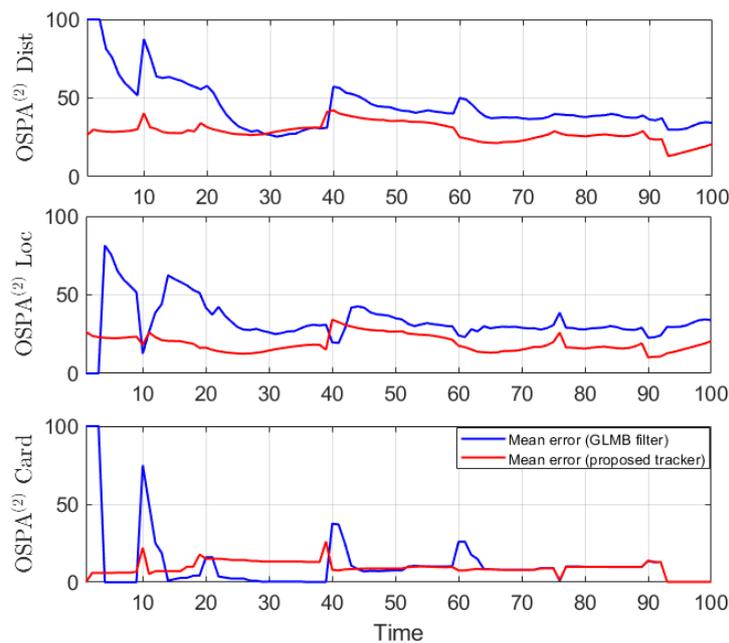


Figure 7. OSPA² error for nonlinear dynamic scenario.

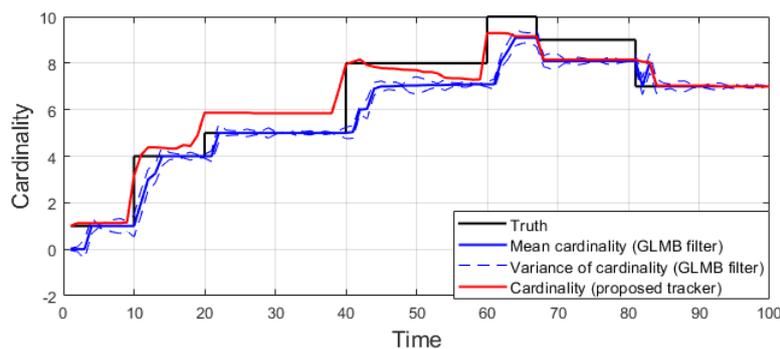


Figure 8. Estimated cardinality for nonlinear dynamic scenario.

The dynamic model and standard observation model are similar to the ones in Section 4.1.1 with $\sigma_v = 1$ pixel/s, $p_S = 0.98$, and $\sigma_\varepsilon = 4$ pixels with a clutter rate of 10. The expected new births states are $m_B^{(1)} = [5, 0, 25, 0]^T$, $m_B^{(2)} = [5, 0, 90, 0]^T$, $m_B^{(3)} = [80, 0, 90, 0]^T$, $m_B^{(4)} = [5, 0, 5, 0]^T$, and $m_B^{(5)} = [90, 0, 30, 0]^T$ with the covariance of $P_B = \text{diag}([3, 2, 3, 2])$ and the probability r_B of 0.03. The ground truth location of objects is shown in Figure 9 while Figure 10 shows samples of raw image observation along with points detection. The implementation of the filtering phase is as the same as in Reference [27]. The smoothing interval is set to the entire tracking time with the track pruning threshold of the smoother set to 3 time steps.

This experiment is run over 100 Monte Carlo trials. The means of OSPA error and OSPA² error are shown respectively in Figures 11 and 12. The estimated cardinality is plotted in Figure 13.

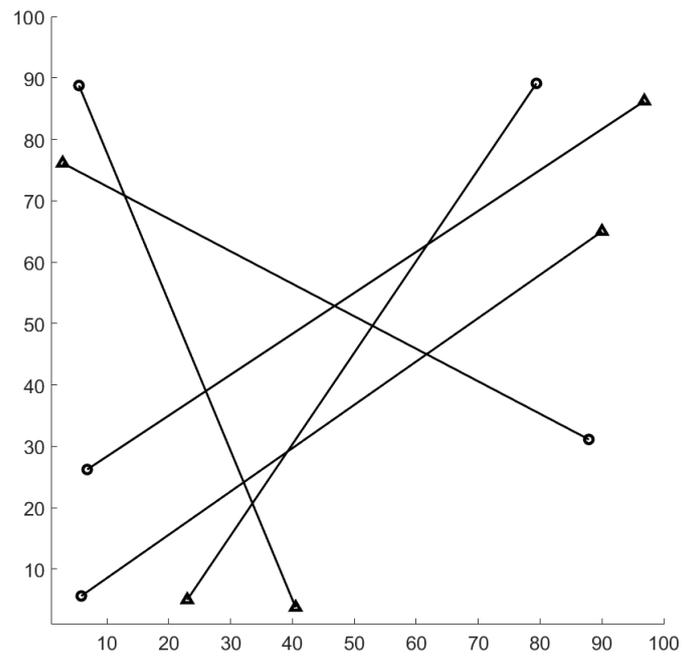


Figure 9. Ground truth for a hybrid track-before-detect (TBD) scenario (circle: track start position, triangle: track end position).

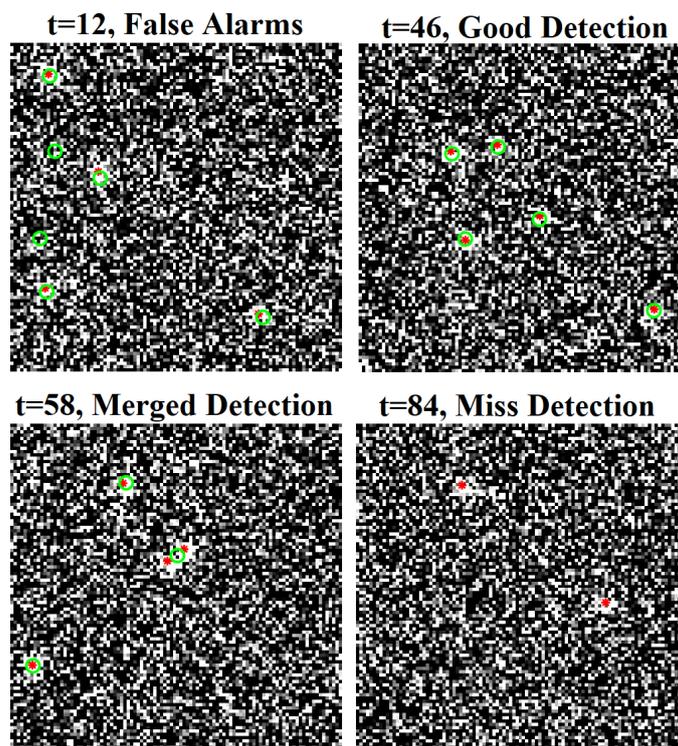


Figure 10. Samples of raw images and point observations for a hybrid TBD scenario (red asterisk: ground truth position, green circle: point detection).

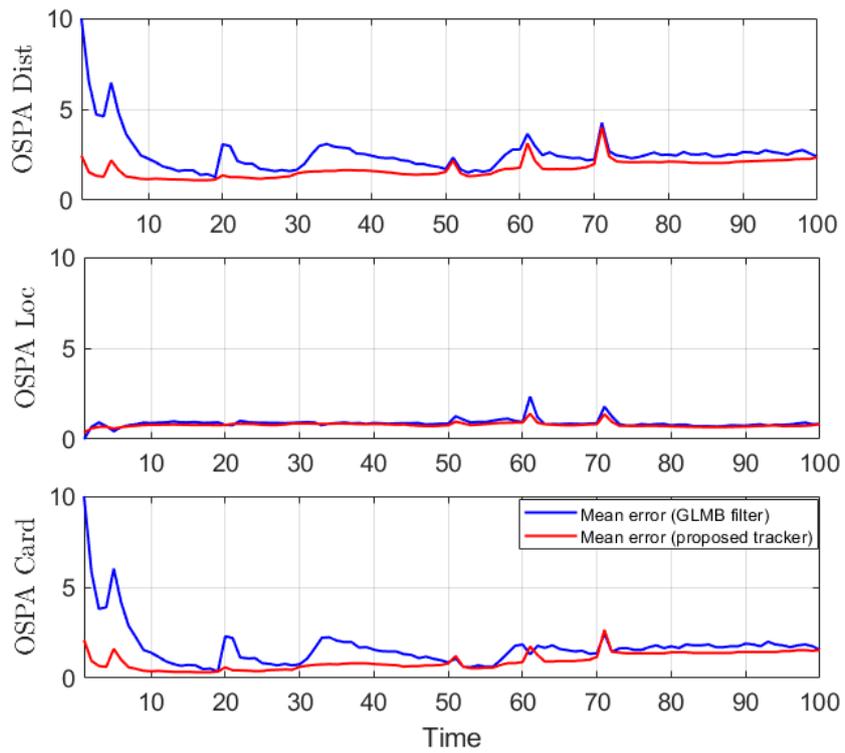


Figure 11. OSPA error for a hybrid TBD scenario.

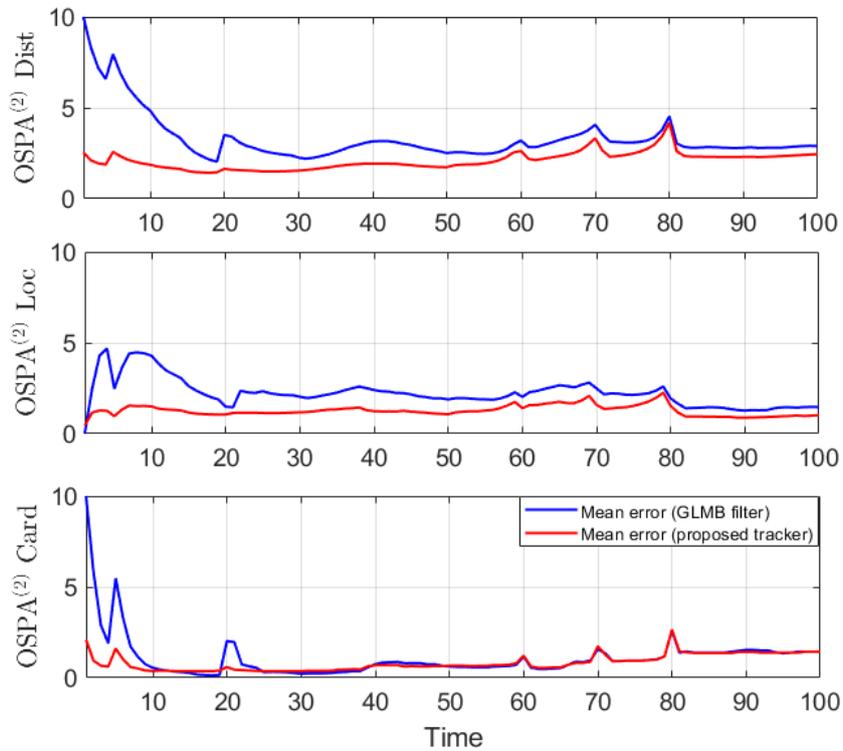


Figure 12. OSPA² error for a hybrid TBD scenario.

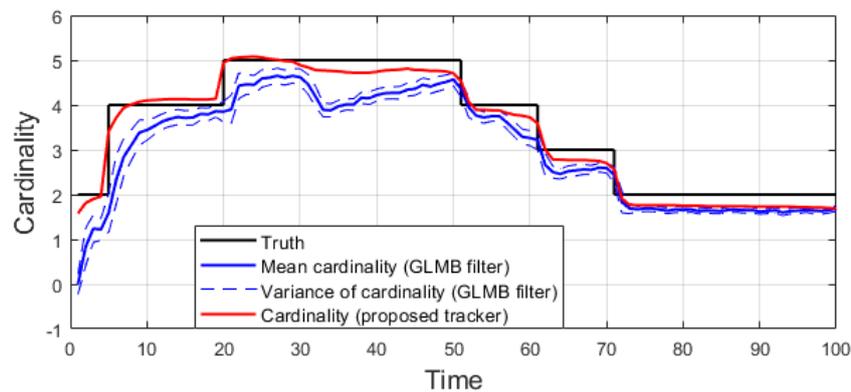


Figure 13. Estimated cardinality for a hybrid TBD scenario.

4.1.4. Discussion on the Simulation Results

For all simulated experiments, we observe lower OSPA and OSPA² errors for the proposed tracker compared to the GLMB filter results. In the first two experiments with the standard observation model, as the clutter rate is high, the filtered-only trajectories jiggle around the true paths due to false measurements. In Figures 2 and 6 as well as in Figures 3 and 7, the overall errors of the GLMB filter estimates are higher than of the proposed tracker estimates. The reduction of localization error contribute mainly to the improvement of the tracking performance. From the cardinality plots in Figures 4 and 8, on average, the proposed tracker slightly improves estimate cardinality performance as it is able to eliminate track fragmentation while eliminating incorrect tracks at some time steps.

In the hybrid TBD tracking experiment, as tracks are miss-detected due to low SNR, the proposed tracker improves tracking performance by eliminating track fragmentation. Not much localization error is reduced by the smoother step as the GLMB filter produces relatively good tracking results. The OSPA and OSPA² results presented in Figures 11 and 12 show slight improvement of the proposed tracker results compared to GLMB filter tracking results. However, the cardinality plot in Figure 13 clearly indicates that the proposed tracker is able to improve the estimated cardinality between time step 30 and 40.

The run time for all simulated scenario is given in Figure 14 in terms of the percentage of extra computational time of the proposed tracker over the computational time of the filtering step only. It is shown that the extra computational time is negligible in all three tracking scenarios with the extra computational time of the proposed tracker less than 0.5% of the filtering computational time. However, the main disadvantage is that the tracker needs to wait until the end of the smoothing interval to be able to produce tracking results.

4.2. Application to Cell Microscopy

In this experiment, we attempt to track biological cells from a sequence of images containing 90 frames by using the proposed tracker. A snapshot of the sequence is shown in Figure 15. In this application, we use the constant turn rate for the dynamic model as in Section 4.1.2 and the standard observation model as in Section 4.1.1. We also implement the measurement driven model as described in Reference [20]. For the first time step, the birth rate is set to a very high value (≈ 1) to initialize objects. Subsequently, the birth rate is capped at 10^{-7} . The standard deviation of the turn rate noise is $\pi/90$ rad/s, and the standard deviation of the velocity noise is 5 pixels/frame. The number of hypotheses is capped at 10,000. The detection rate is set to 0.88, and the surviving rate and the spawning rate are 0.999 and 0.035, respectively. The clutter rate is set to 0.05. The cell spawning model is the same as described in Reference [57] with the covariance of the spawning model given

as $Q_T = \begin{bmatrix} 40 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & \pi/90 \end{bmatrix}$ and the smoothing interval set to the entire image sequence. In this application, we set the track pruning threshold of the estimator to 3 time steps.

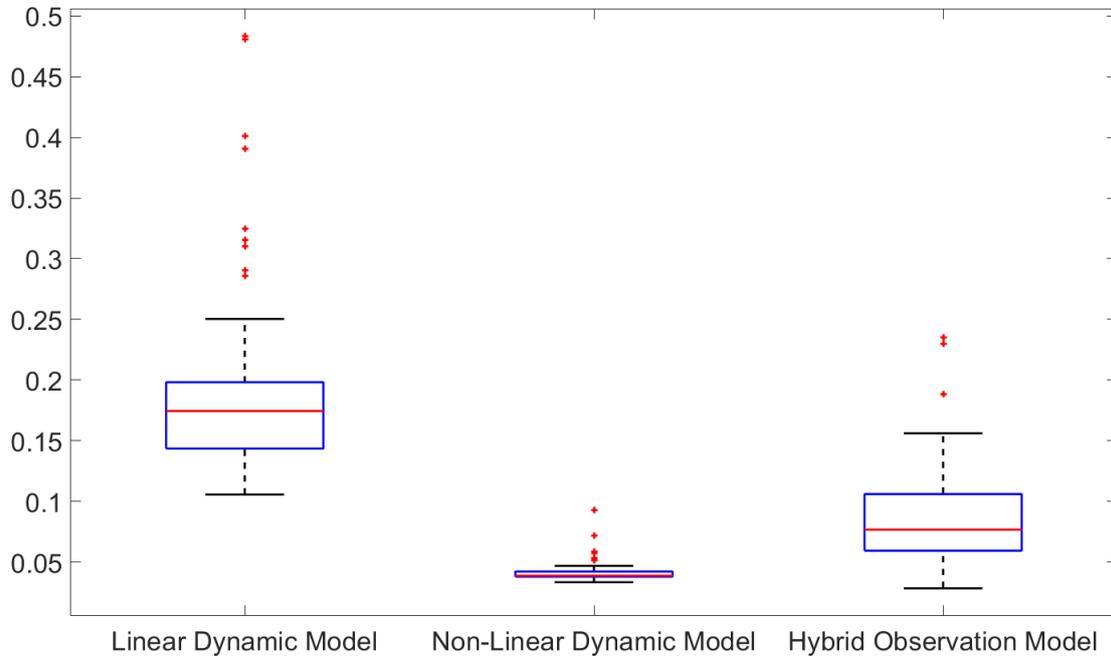


Figure 14. Percentage of smoothing time over filtering time.

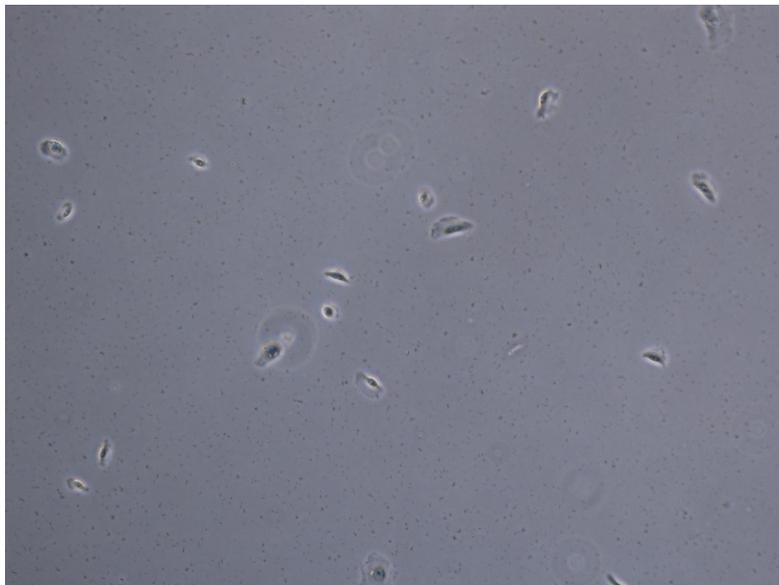


Figure 15. Snapshot of biological cell sequence.

From the tracking results, significant improvement is observed as the proposed tracker is able to eliminate incorrect spawned tracks. While the OSPA error in Figure 16 shows similar performance for the GLMB filter and the proposed tracker, the improvement is clearly reflected in the OSPA² cardinality error plots in Figure 17. From the cardinality plot in Figure 18, the estimated cardinality from our tracker is much closer to the true values as fewer incorrect spawned tracks are estimated. In this

experiment, there is not much difference between the GLMB filter and the proposed tracker estimates localization error due to the mismatch between the dynamic model and actual motion of the cells. Finally, in Figure 19, we illustrate the improved tracking results in terms of tracking sequence for several time steps at a selected region where the cell splitting process occurs.

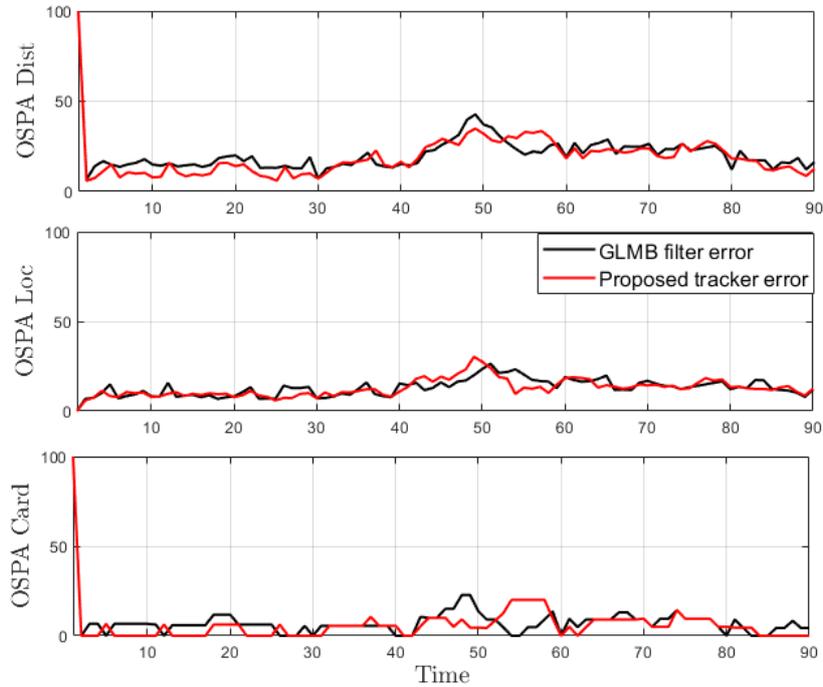


Figure 16. OSPA error for tracking biological cells.

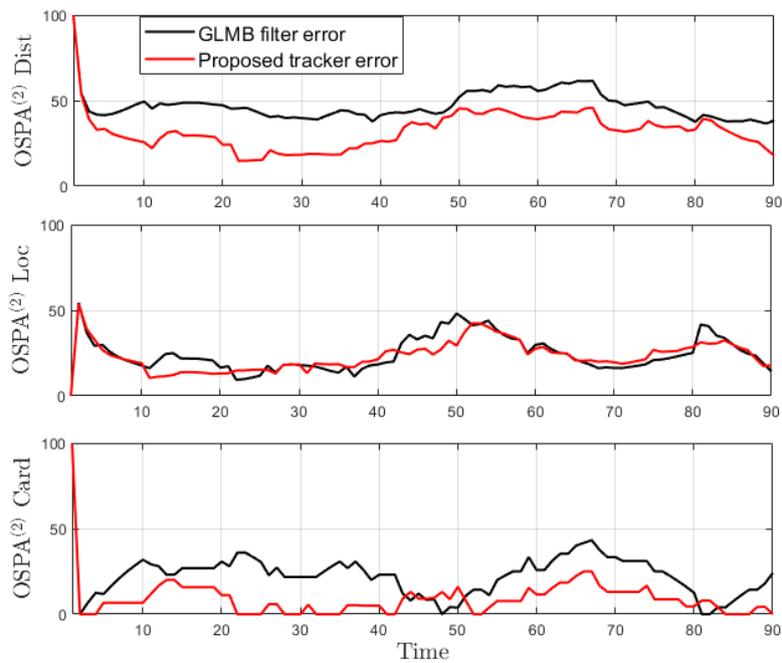


Figure 17. OSPA² error for tracking biological cells.

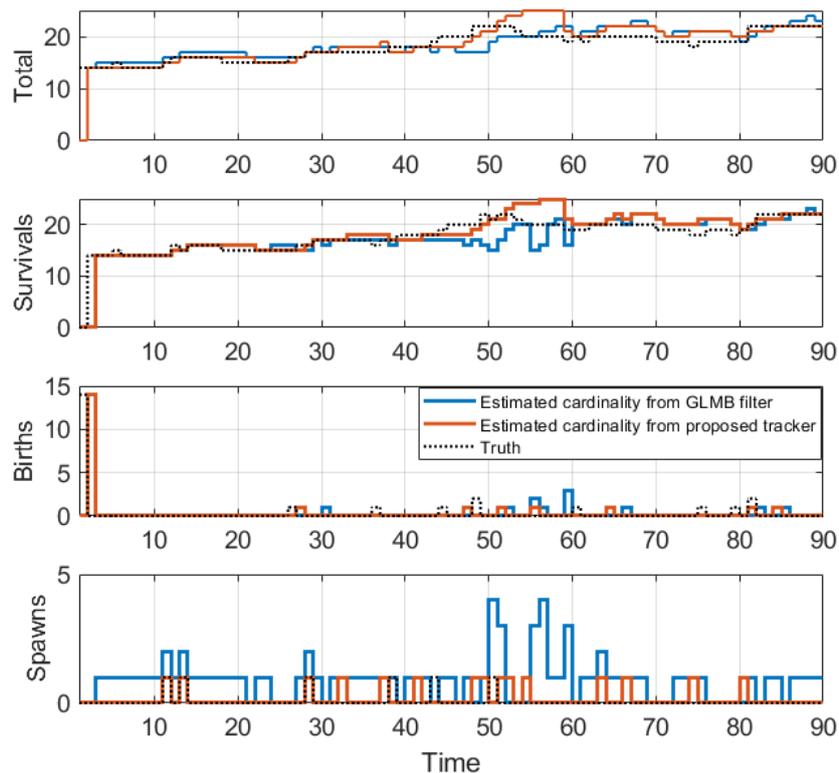


Figure 18. Estimated cardinality for tracking biological cells.

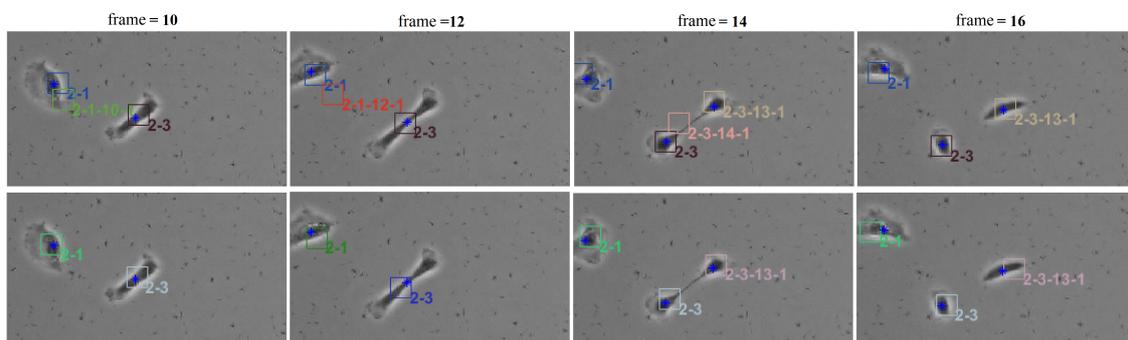


Figure 19. The tracked image sequences of biological cells with blue asterisks denoting points detection. Top row: Generalized Labeled Multi-Bernoulli (GLMB) filter tracking results. Bottom: Proposed tracker tracking results.

5. Conclusions

In this paper, we detailed the implementation of a new tracker based on GLMB filter and a modified multi-scan estimator. In addition to lowering the localization error by performing RTS smoother on each individual estimated trajectory, the proposed tracker can also reduce cardinality errors by deleting the short-term tracks via track management and by completely eliminating track fragmentation. The computation time is shown to contribute to less than 0.5% of the total tracking time, although a fixed delay time is needed before the tracker can produce the estimate. Therefore, in applications when real-time updates are not required, the proposed tracker can be used to improve the tracking results given negligible extra computation time. However, as the smoothing results strongly depend on the quality of the estimates obtained from the forward filtering step, if the filtered estimate experiences strong distortion, the performance of the proposed tracker degrades significantly.

Author Contributions: Methodology, T.T.D.N. and D.Y.K.; visualization, T.T.D.N.; writing—original draft, T.T.D.N.; writing—review and editing, T.T.D.N. and D.Y.K.

Funding: This work was funded by the Australian Research Council grant number DP160104662.

Acknowledgments: The authors acknowledge the administrative and technical support from Curtin University and RMIT University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fortmann, T.; Bar-Shalom, Y.; Scheffe, M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Ocean. Eng.* **1983**, *8*, 173–184. [[CrossRef](#)]
2. Reid, D. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* **1979**, *24*, 843–854. [[CrossRef](#)]
3. Vo, B.N.; Vo, B.T.; Mahler, R.P.S. Closed form solutions to forward-backward smoothing. *IEEE Trans. Signal Process.* **2012**, *60*, 2–17. [[CrossRef](#)]
4. Vo, B.N.; Ma, W.K. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Trans. Signal Process.* **2006**, *54*, 4091–4104. [[CrossRef](#)]
5. Gao, Y.; Jiang, D.; Liu, M. Particle-Gating SMC-PHD filter. *Signal Process.* **2017**, *130*, 64–73. [[CrossRef](#)]
6. Wang, S.; Bao, Q.; Chen, Z. Refined PHD Filter for Multi-Target Tracking under Low Detection Probability. *Sensors* **2019**, *19*, 2842. [[CrossRef](#)] [[PubMed](#)]
7. Zheng, J.; Gao, M. Tracking Ground Targets with a Road Constraint Using a GMPHD Filter. *Sensors* **2018**, *18*, 2723. [[CrossRef](#)]
8. Vo, B.T.; Vo, B.N.; Cantoni, A. Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter. *IEEE Trans. Signal Process.* **2007**, *55*, 3553–3567. [[CrossRef](#)]
9. Zhang, H.; Jing, Z.; Hu, S. Gaussian mixture CPHD filter with gating technique. *Signal Process.* **2009**, *89*, 1521–1530. [[CrossRef](#)]
10. Si, W.; Wang, L.; Qu, Z. Multi-Target Tracking Using an Improved Gaussian Mixture CPHD Filter. *Sensors* **2016**, *16*, 1964. [[CrossRef](#)]
11. Vo, B.T.; Vo, B.N.; Cantoni, A. The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and Its Implementations. *IEEE Trans. Signal Process.* **2009**, *57*, 409–423. [[CrossRef](#)]
12. Hu, X.; Ji, H.; Liu, L. Adaptive Target Birth Intensity Multi-Bernoulli Filter with Noise-Based Threshold. *Sensors* **2019**, *19*, 1120. [[CrossRef](#)] [[PubMed](#)]
13. Vo, B.T.; Vo, B.N. Labeled Random Finite Sets and Multi-Object Conjugate Priors. *IEEE Trans. Signal Process.* **2013**, *61*, 3460–3475. [[CrossRef](#)]
14. Yang, B.; Wang, J.; Wang, W. An efficient approximate implementation for labeled random finite set filtering. *Signal Process.* **2018**, *150*, 215–227. [[CrossRef](#)]
15. Vo, B.N.; Vo, B.T.; Phung, D. Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter. *IEEE Trans. Signal Process.* **2014**, *62*, 6554–6567. [[CrossRef](#)]
16. Vo, B.N.; Vo, B.T.; Hoang, H.G. An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter. *IEEE Trans. Signal Process.* **2017**, *65*, 1975–1987. [[CrossRef](#)]
17. Moratuwage, D.; Adams, M.; Inostroza, F. δ -Generalized Labeled Multi-Bernoulli Simultaneous Localization and Mapping with an Optimal Kernel-Based Particle Filtering Approach. *Sensors* **2019**, *19*, 2290. [[CrossRef](#)] [[PubMed](#)]
18. Liu, C.; Sun, J.; Lei, P.; Qi, Y. δ -Generalized Labeled Multi-Bernoulli Filter Using Amplitude Information of Neighboring Cells. *Sensors* **2018**, *18*, 1153. [[CrossRef](#)]
19. Do, C.T.; Van Nguyen, H. Tracking Multiple Targets from Multistatic Doppler Radar with Unknown Probability of Detection. *Sensors* **2019**, *19*, 1672. [[CrossRef](#)]
20. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. The Labeled Multi-Bernoulli Filter. *IEEE Trans. Signal Process.* **2014**, *62*, 3246–3260. [[CrossRef](#)]
21. Xie, Y.; Song, T.L. Bearings-only multi-target tracking using an improved labeled multi-Bernoulli filter. *Signal Process.* **2018**, *151*, 32–44. [[CrossRef](#)]
22. Amrouche, N.; Khenchaf, A.; Berkani, D. Multiple target tracking using track before detect algorithm. In Proceedings of the 2017 International Conference on Electromagnetics in Advanced Applications (ICEAA), Verona, Italy, 11–15 September 2017; pp. 692–695. [[CrossRef](#)]
23. Grossi, E.; Lops, M.; Venturino, L. A Track-Before-Detect Algorithm with Thresholded Observations and Closely-Spaced Targets. *IEEE Signal Process. Lett.* **2013**, *20*, 1171–1174. [[CrossRef](#)]

24. Yi, W.; Morelande, M.R.; Kong, L.; Yang, J. An Efficient Multi-Frame Track-before-Detect Algorithm for Multi-Target Tracking. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 421–434. [[CrossRef](#)]
25. Rathnayake, T.; Tennakoon, R.; Khodadadian Gostar, A.; Bab-Hadiashar, A.; Hoseinnezhad, R. Information Fusion for Industrial Mobile Platform Safety via Track-before-Detect Labeled Multi-Bernoulli Filter. *Sensors* **2019**, *19*, 2016. [[CrossRef](#)] [[PubMed](#)]
26. Vo, B.N.; Vo, B.T.; Pham, N.T.; Suter, D. Joint Detection and Estimation of Multiple Objects from Image Observations. *IEEE Trans. Signal Process.* **2010**, *58*, 5129–5141. [[CrossRef](#)]
27. Kim, D.Y.; Vo, B.N.; Vo, B.T.; Jeon, M. A labeled random finite set online multi-object tracker for video data. *Pattern Recognit.* **2019**, *90*, 377–389. [[CrossRef](#)]
28. Bryant, D.S.; Delande, E.D.; Gehly, S.; Houssineau, J.; Clark, D.E.; Jones, B.A. The CPHD Filter with Target Spawning. *IEEE Trans. Signal Process.* **2017**, *65*, 13124–13138. [[CrossRef](#)]
29. Lundgren, M.; Svensson, L.; Hammarstrand, L. A CPHD Filter for Tracking with Spawning Models. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 496–507. [[CrossRef](#)]
30. Bryant, D.S.; Vo, B.; Jones, B.A. A Generalized Labeled Multi-Bernoulli Filter with Object Spawning. *IEEE Trans. Signal Process.* **2018**, *66*, 6177–6189. [[CrossRef](#)]
31. Bryson, A.; Frazier, M. *Smoothing for Linear and Nonlinear Dynamic Systems*; Tech. Rep. TDR-63-119; Aeronautical Systems Division Wright Patterson AFB: Dayton, OH, USA, 1962.
32. Rauch, H.E.; Striebel, C.T.; Tung, F. Maximum likelihood estimates of linear dynamic systems. *AIAA J.* **1965**, *3*, 1445–1450. [[CrossRef](#)]
33. Fraser, D.; Potter, J. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Trans. Autom. Control* **1969**, *14*, 387–390. [[CrossRef](#)]
34. Briers, M.; Doucet, A.; Maskell, S. Smoothing algorithms for state-space models. *Ann. Inst. Stat. Math.* **2009**, *62*, 61–89. [[CrossRef](#)]
35. Fearnhead, P.; Wyncoll, D.; Tawn, J. A sequential smoothing algorithm with linear computational cost. *Biometrika* **2010**, *97*, 447–464. [[CrossRef](#)]
36. Särkkä, S. Unscented Rauch–Tung–Striebel Smoother. *IEEE Trans. Autom. Control* **2008**, *53*, 845–849. [[CrossRef](#)]
37. Särkkä, S. Continuous-time and continuous–discrete-time unscented Rauch–Tung–Striebel smoothers. *Signal Process.* **2010**, *90*, 225–235. [[CrossRef](#)]
38. Kitagawa, G. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *J. Comput. Graph. Stat.* **1996**, *5*, 1–25. [[CrossRef](#)]
39. Simon, J.; Julier, J.K.U. New extension of the Kalman filter to nonlinear systems. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 21–24 April 1997; Volume 3068. [[CrossRef](#)]
40. Chen, B.; Tugnait, J. Multisensor tracking of a maneuvering target in clutter using IMM-PDA fixed-lag smoothing. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 983–991. [[CrossRef](#)]
41. Helmick, R.; Blair, W.; Hoffman, S. One-step fixed-lag smoothers for Markovian switching systems. *IEEE Trans. Autom. Control* **1996**, *41*, 1051–1056. [[CrossRef](#)]
42. Li, T.; Chen, H.; Sun, S.; Corchado, J.M. Joint Smoothing and Tracking Based on Continuous-Time Target Trajectory Function Fitting. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1476–1483. [[CrossRef](#)]
43. Mahler, R.P.S.; Vo, B.T.; Vo, B.N. Forward-Backward Probability Hypothesis Density Smoothing. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 707–728. [[CrossRef](#)]
44. He, X.; Liu, G. Improved Gaussian mixture probability hypothesis density smoother. *Signal Process.* **2016**, *120*, 56–63. [[CrossRef](#)]
45. Nadarajah, N.; Kirubarajan, T.; Lang, T.; McDonald, M.; Punithakumar, K. Multitarget Tracking Using Probability Hypothesis Density Smoothing. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 2344–2360. [[CrossRef](#)]
46. Nagappa, S.; Delande, E.D.; Clark, D.E.; Houssineau, J. A Tractable Forward-Backward CPHD Smoother. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 201–217. [[CrossRef](#)]
47. Li, D.; Hou, C.; Yi, D. Multi-Bernoulli smoother for multi-target tracking. *Aerosp. Sci. Technol.* **2016**, *48*, 234–245. [[CrossRef](#)]
48. Koch, W. Fixed-interval retrodiction approach to Bayesian IMM-MHT for maneuvering multiple targets. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 2–14. [[CrossRef](#)]

49. Beard, M.; Vo, B.T.; Vo, B.N. Generalised labeled multi-Bernoulli forward-backward smoothing. In Proceedings of the 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 688–694.
50. Vo, B.N.; Vo, B.T. A Multi-Scan Labeled Random Finite Set Model for Multi-Object State Estimation. *IEEE Trans. Signal Process.* **2019**, *67*, 4948–4963. [[CrossRef](#)]
51. Mahler, R. Exact Closed-Form Multitarget Bayes Filters. *Sensors* **2019**, *19*, 818. [[CrossRef](#)]
52. Beard, M.; Vo, B.T.; Vo, B.N. A Solution for Large-Scale Multi-Object Tracking. *arXiv* **2018**, arXiv:1804.06622.
53. Vo, B.N.; Vo, B.T.; Beard, M. Multi-Sensor Multi-Object Tracking with the Generalized Labeled Multi-Bernoulli Filter. *IEEE Trans. Signal Process.* **2019**. [[CrossRef](#)]
54. Mahler, R.P.S. *Statistical Multisource-Multitarget Information Fusion*; Artech House: Norwood, MA, USA, 2007.
55. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457. [[CrossRef](#)]
56. Beard, M.; Vo, B.T.; Vo, B.N. OSPA(2): Using the OSPA metric to evaluate multi-target tracking performance. In Proceedings of the 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 31 October–1 November 2017; pp. 86–91. [[CrossRef](#)]
57. Nguyen, T.T.D.; Kim, D.Y. On-line Tracking of Cells and Their Lineage from Time Lapse Video Data. In Proceedings of the 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 24–27 October 2018; pp. 291–296. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).