A Solution for Large-Scale Multi-Object Tracking

Michael Beard, Ba Tuong Vo, and Ba-Ngu Vo

***Abstract*—A large-scale multi-object tracker based on the generalised labeled multi-Bernoulli (GLMB) filter is proposed. The algorithm is capable of tracking a very large, unknown and time-varying number of objects simultaneously, in the presence of a high number of false alarms, as well as missed detections and measurement origin uncertainty due to closely spaced objects. The algorithm is demonstrated on a simulated tracking scenario, where the peak number objects appearing simultaneously exceeds one million. Additionally, we introduce a new method of applying the optimal sub-pattern assignment (OSPA) metric to determine a meaningful distance between two sets of tracks. We also develop an efficient strategy for its exact computation in large-scale scenarios to evaluate the performance of the proposed tracker.**

***Index Terms*—Random finite sets, generalised labeled multi-Bernoulli, multi-object tracking, large-scale tracking, OSPA.**

I. Introduction

MULTI-OBJECT tracking is a problem with a wide variety of applications across diverse disciplines, and numerous effective solutions have been developed in recent decades [1]–[3]. The common goal of multi-object tracking is to estimate the trajectories of an unknown and time-varying number of objects, using sensor measurements corrupted by phenomena including observation noise, false alarms, missed detections, and data association uncertainty. The combination of these effects gives rise to a highly demanding computational task, with complexity that grows exponentially as the number of objects/measurements increases. Tracking a very large number of objects simultaneously (in excess of hundreds of thousands) is thus a challenging problem, with important practical applications. A few notable examples are: (i) space situational awareness, which requires tracking thousands of satellites and millions of debris objects [4]–[6]; (ii) wide area surveillance (e.g. monitoring large urban environments), requires tracking hundreds of thousands of objects over time, including vehicles and people in crowded environments [7]–[9]; (iii) cell biology, where tracking the motion of large numbers of cells is critical to understanding their behaviour in living tissues [10], [11]; (iv) wildlife biology, where tracking large animal populations is needed to study the behaviour of wildlife in their natural habitats [16].

Manuscript received July 13, 2019; revised February 13, 2020; accepted March 29, 2020. Date of publication April 10, 2020; date of current version June 5, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Youngchul Sung. This work was supported by the Australian Research Council under Discovery Projects DP170104584 and DP160104662. *(Corresponding author: Ba Tuong Vo.)*

The authors are with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, WA 6102, Australia (e-mail: michael.beard@curtin.edu.au; ba-tuong.vo@curtin.edu.au; ba-ngu.vo@curtin.edu.au).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors. This includes two illustrative videos of the tracking results.

Digital Object Identifier 10.1109/TSP.2020.2986136

The total number of objects in a scenario is often quoted as a major factor influencing the computational complexity of a multi-object tracking problem. This is partially true, since an increase in the number of objects leads to an increase in the number of possible events that a tracking algorithm must consider. However, this is not the only concern, and a more meaningful indication of complexity is the density of the objects and measurements in both space and time. For example, it is straightforward to track a large cumulative number of objects over time, when only a small number are present simultaneously at any given instant. Likewise, a large number of spatially well-separated objects is relatively easy to track, since they can be considered as statistically independent entities. In these cases, it is likely that running single-object filters in parallel would suffice. Difficulties begin to arise when objects come into close spatial proximity from the point of view of the sensor, and the ensuing increase in data association ambiguity leads to a combinatorial explosion in the number of statistically likely observation events. In dynamic multi-object scenarios, the problem is further compounded by the motion of the objects.

Various methods have been proposed to address the computational challenges of tracking a large number of objects simultaneously. One of the earliest works on large scale multi-object tracking proposed the use of efficient spatial searching algorithms to associate measurements to tracks [12]. Although capable of processing a very large number of objects, the proposed method does not account for dependencies between closely spaced objects, and thus its performance is likely to degrade in such cases. In [13] it was suggested that the same techniques could be applied to help improve the efficiency of the joint probabilistic data association (JPDA) algorithm, but no numerical results were provided. An alternative approach was proposed in [14], known as linear multi-target integrated probabilistic data association (LMIPDA). This method reduces computation using an approximation that treats nearby objects as additional sources of clutter, and the resulting algorithm was demonstrated on a simulated 50-object scenario. An approach based on 2D assignment of measurements to tracks was proposed in [15], with an application to large-scale air traffic surveillance. The scenario had a total of about 800 tracks, however, the target/measurement density and the number of simultaneous objects was not provided.

Algorithms based on multiple hypothesis tracking (MHT) have also been proposed for tracking large numbers of objects. For example, applications to cell tracking [10] and wildlife tracking [16], have been demonstrated on thousands of objects in total, with several hundred objects simultaneously [16]. The labelled multi-Bernoulli (LMB) filter, a one-term approximation of the generalised labeled multi-Bernoulli (GLMB) filter, has also been demonstrated on a simulated scenario with over a thousand objects simultaneously [17], and in [18] it was used with spatial searching to track hundreds of sea-ice objects.

An alternative approach to improving the scalability of multi-target tracking systems is the concept of distributed estimation, which spreads the computational load across multiple sensor nodes, each of which only processes observations from its local region. Random finite set (RFS) based multi-sensor fusion algorithms have been proposed for the PHD/CPHD filters [19]–[22], multi-Bernoulli filter [23]–[26], and hybrid Poisson multi-Bernoulli filter [27]. However these approaches are not true multi-object trackers, since only the current states are estimated. Multi-sensor fusion for multi-object tracking filters was first examined in [28], where fusion rules for the LMB filter [29] and marginalised GLMB filter [30], were proposed and demonstrated for multi-sensor multi-object tracking. A variation for the LMB filter was subsequently proposed in [31], where sensor fusion is performed based on a Cauchy–Schwarz divergence. Further works on robustness of distributed multi-object estimation have been reported in [32], [33], and the latest works on computationally efficient implementations were proposed in [34].

To satisfy the competing demands of computational efficiency versus accuracy, a trade-off is necessary. In this paper, we present a multi-object tracking filter that accomplishes this trade-off by exploiting the properties of GLMB densities, and the standard multi-object likelihood through a principled approximation. The result is an algorithm that can be applied to large-scale multi-object tracking scenarios exhibiting commonly encountered structural properties, without suffering from an intractable increase in computational complexity. The algorithm is highly effective when the scenario consists of isolated groups of high object/measurement density, and is capable of adapting to changes in the structure of these groups over time.

Our proposed method is based on functional approximation of the multi-object density (equivalent to a probability density for finite-set-valued random variable [35]) – a key element of the random finite set (RFS) approach – that encapsulates all information on the current set of tracks in a single non-negative function. Processing the large number of combinations of events translates to recursive computation of the multi-object filtering density [36]–[38]. Tractability hinges on efficient functional approximation/computation of the so-called GLMB filtering recursion, under limited processing/memory resources. Conceptually, the key enablers in our proposed large-scale tracker are: (i) adaptive approximation of the GLMB filtering density, at each time, by a product of tractable and approximately independent GLMB densities; and (ii) efficient parallel computation of these GLMB densities by exploiting the conjugacy of the GLMB family. This strategy is distinct from the approach in [17], where the GLMB is approximated by a single term. In essence, our strategy efficiently identifies and processes significant combinations by exploiting structural properties and parallelisation, to make the most of the limited computing resources. Consequently, while the focus of this paper is on large-scale problems, our solution also provides significant efficiency gains when applied to smaller scale problems.

Our study would be incomplete without evaluating the tracking performance of the proposed multi-object tracker, which is a challenging task in itself [39], [40]. We require a measure of dissimilarity between two sets of tracks, which: (i) is physically meaningful; (ii) satisfies the properties of a metric for mathematical consistency; and (iii) is computable for scenarios involving millions of tracks. The optimal sub-pattern assignment (OSPA) metric [41], in its most commonly used form, measures the distance between two sets of states, and does not take into account phenomena such as track switching and track fragmentation. Nonetheless, by developing a meaningful base-distance between two tracks, we are able to use OSPA to construct a physically meaningful distance between two sets of tracks, which we called OSPA$^{(2)}$ to distinguish it from the standard use. To evaluate the performance of the proposed tracker on a scenario involving an unknown and time-varying number of objects with a peak in excess of one million, we developed a scalable procedure for exact computation of the OSPA$^{(2)}$ metric. Preliminary results on OSPA$^{(2)}$ were published in [42], [43]. This paper provides complete mathematical details.

The rest of the paper is structured as follows. Section II provides the necessary background on the GLMB filter. Section III presents some theoretical results regarding the decomposition of GLMB densities. In section IV we apply these results to implement an efficient GLMB filter, capable of handling large-scale multi-object tracking problems. Sections V and VI present the OSPA$^{(2)}$ metric, and its use to evaluate the proposed large-scale tracker. Some concluding remarks are given in section VII. Mathematical proofs are given in the Appendix.

II. Background: Generalised Labeled Multi-Bernoulli Tracker

In multi-object systems, *tracking* is distinct from *filtering*, in the sense that tracking involves the estimation of the trajectories of objects over time, as opposed to the multi-object state at each time instant. The generalised labelled multi-Bernoulli (GLMB) filter is an algorithm that is specifically designed to provide estimates of object trajectories by modeling the multi-object state as a *labeled* random finite set (RFS). In this section we briefly revisit the GLMB filter, and the interested reader is referred to [36]–[38] for more detailed treatments.

We begin by defining the notion of a labeled RFS. Let $\mathbb{X}$ be a single-object state space, $\mathbb{L}$ a discrete label space, $\mathcal{L}: \mathbb{X} \times \mathbb{L} \to \mathbb{L}$ the projection defined by $\mathcal{L}((x,\ell)) = \ell$ for all points $(x,\ell) \in \mathbb{X} \times \mathbb{L}$. Denote by $\mathcal{F}(S)$ the collection of all finite subsets of some underlying space S. Now consider $\boldsymbol{X} \in \mathcal{F}(\mathbb{X} \times \mathbb{L})$ and its corresponding label set $\mathcal{L}(\boldsymbol{X}) = \{\mathcal{L}(\boldsymbol{x}) : x \in \boldsymbol{X}\}$. Then the labels of the points in $\boldsymbol{X}$ are distinct if and only if $\boldsymbol{X}$ and its label set $\mathcal{L}(\boldsymbol{X})$ have equal cardinality. This is expressed mathematically by defining a *distinct label indicator* function

$$\Delta(\boldsymbol{X}) \triangleq \delta_{|\boldsymbol{X}|}[|\mathcal{L}(\boldsymbol{X})|],$$

which has value 1 if the labels in $\boldsymbol{X}$ are distinct and 0 otherwise. A labeled RFS is defined as a marked RFS with distinct marks [36]. More precisely, a labeled RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ is an RFS of $\mathbb{X} \times \mathbb{L}$, constructed by marking the elements of an RFS of $\mathbb{X}$ with distinct labels from $\mathbb{L}$, i.e. any realisation $\boldsymbol{X}$ must satisfy $\Delta(\boldsymbol{X}) = 1$.

A. Multi-Object Dynamic Model

Given the multi-object state $\boldsymbol{X}$ (at time k) with label space $\mathbb{L}$, each $(x,\ell) \in \boldsymbol{X}$ either survives with probability $P_S(x,\ell)$ and evolves to a new state (x_+,ℓ_+) (at time $k+1$) with probability density $f_+(x_+|x,\ell)\delta_\ell[\ell_+]$ or dies with probability $1 - P_S(x,\ell)$. The surviving label space $\mathbb{L} \triangleq \mathbb{L}_k$ is given by a disjoint union of birth label spaces $\mathbb{B}_t$ for all times $t = 0,\ldots,k$, i.e. $\mathbb{L}_k = \biguplus_{t=0}^{k} \mathbb{B}_t$. To ensure that the birth label spaces are disjoint,

each birth label is constructed as an ordered pair, consisting of the birth time and a unique index, i.e. $\mathbb{B}_t = \{(t,i)\}_{i=1}^{|\mathbb{B}_t|}$. The set $\boldsymbol{B}_+$ of new objects (born at time $k+1$) with birth label space $\mathbb{B}_+ \triangleq \mathbb{B}_{k+1}$ is distributed according to the labeled multi-Bernoulli (LMB) density

$$\boldsymbol{f}_{B,+}^{(\mathbb{B}_+)}(\boldsymbol{B}_+) = \Delta(\boldsymbol{B}_+)\left[1_{\mathbb{B}_+} r_{B,+}\right]^{\mathcal{L}(\boldsymbol{B}_+)}\left[1-r_{B,+}\right]^{\mathbb{B}_+ - \mathcal{L}(\boldsymbol{B}_+)} p_{B,+}^{\boldsymbol{B}_+},$$

where $[h]^X \triangleq \prod_{x\in X} h(x)$ (with $[h]^{\emptyset}=1$) is a *multi-object exponential,* $r_{B,+}(\ell)$ is the probability that a new object with label ℓ is born, and $p_{B,+}(\cdot,\ell)$ is the distribution of its kinematic state [36]. The multi-object state $\boldsymbol{X}_+$ (at time $k+1$) with label space $\mathbb{L}_+ \triangleq \mathbb{L}\cup\mathbb{B}_+$ is formed by the union of surviving objects and new born objects. Using the standard assumption that, conditional on $\boldsymbol{X}$, objects move, appear and die independently of each other, the expression for the multi-object transition density $\boldsymbol{f}_+$ is given by [36]–[38]

$$\boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X}) = \boldsymbol{f}_{S,+}(\boldsymbol{X}_+\cap(\mathbb{X}\times\mathbb{L})\,|\boldsymbol{X})\,\boldsymbol{f}_{B,+}^{(\mathbb{B}_+)}(\boldsymbol{X}_+-(\mathbb{X}\times\mathbb{L}))$$

where

$$\begin{aligned}\boldsymbol{f}_{S,+}(\boldsymbol{W}|\boldsymbol{X}) &= \Delta(\boldsymbol{W})\Delta(\boldsymbol{X})1_{\mathcal{L}(\boldsymbol{X})}(\mathcal{L}(\boldsymbol{W}))[\Phi(\boldsymbol{W};\cdot)]^{\boldsymbol{X}}\\ \Phi(\boldsymbol{W};x,\ell) &= \left(1-1_{\mathcal{L}(\boldsymbol{W})}(\ell)\right)(1-P_S(x,\ell))\\ &\quad+\sum_{(x_+,\ell_+)\in\boldsymbol{W}}\delta_\ell[\ell_+]\,P_S(x,\ell)f_+(x_+|x,\ell).\end{aligned}$$

B. Multi-Object Observation Model

For a given multi-object state $\boldsymbol{X}$, each $(x,\ell)\in\boldsymbol{X}$ is either detected with probability $P_D(x,\ell)$ and generates a detection z with likelihood $g(z|x,\ell)$ or missed with probability $1-P_D(x,\ell)$. The *multi-object observation* Z is the union of the observations from detected objects and Poisson clutter with intensity κ. The multi-object likelihood function is given by [36]–[38]

$$g(Z|\boldsymbol{X}) = \sum_{\theta\in\Theta(\mathcal{L}(\mathbf{X}))}\prod_{(x,\ell)\in\boldsymbol{X}}\psi_Z^{(\theta(\ell))}(x,\ell) \tag{1}$$

where Θ is the set of *positive 1-1* maps $\theta:\mathbb{L}\to\{0:|Z|\}$, i.e. maps such that *no two distinct labels are assigned the same positive value*, $\Theta(I)$ is the subset of Θ with domain I, and

$$\psi_{\{z_{1:|Z|}\}}^{(j)}(x,\ell) = \begin{cases}\frac{P_D(x,\ell)g(z_j|x,\ell)}{\kappa(z_j)}, & j\in\{1,\ldots,|Z|\}\\ 1-P_D(x,\ell), & j=0\end{cases}.$$

The map θ specifies that object ℓ generates detection $z_{\theta(\ell)}\in Z$, with $\theta(\ell)=0$ if ℓ is undetected. The positive 1-1 property means that θ is 1-1 on $\{\ell:\theta(\ell)>0\}$, and ensures that any detection in Z is generated from at most one object.

C. Generalised Labelled Multi-Bernoulli Random Finite Sets

A generalised labeled multi-Bernoulli RFS is defined as a class of labeled RFS that is distributed according to a multi-object density with the form

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X})\sum_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}} w^{(I,c)}\delta_I(\mathcal{L}(\boldsymbol{X}))\left[p^{(c)}\right]^{\boldsymbol{X}}, \tag{2}$$

where $\mathcal{F}(\mathbb{L})$ is the space of all finite subsets of $\mathbb{L}$, $\mathbb{C}$ is some finite space, each $w^{(I,\xi)}$ is a non-negative weight such that $\sum_{I\in\mathcal{F}(\mathbb{L})}\sum_{c\in\mathbb{C}} w^{(I,c)}=1$ and each $p^{(c)}(\cdot,\ell)$ is a probability density on $\mathbb{X}$. For convenience, we denote the space of all GLMB densities on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$ by $\mathcal{G}_{\mathbb{L}}$, and adopt the following abbreviated notation for a GLMB density in terms of its parameters

$$\boldsymbol{\pi} \triangleq \left\{\left(w^{(I,c)},p^{(c)}\right)\right\}_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}. \tag{3}$$

If the multi-object filtering density at the current time step is a GLMB given by (3), then the multi-object filtering density at the next time step, given by the multi-object Bayes recursion

$$\boldsymbol{\pi}_+(\boldsymbol{X}_+|Z_+) \propto g(Z_+|\boldsymbol{X}_+)\int\boldsymbol{\pi}(\boldsymbol{X}_+)\,\boldsymbol{f}(\boldsymbol{X}_+|\boldsymbol{X})\delta\boldsymbol{X}, \tag{4}$$

is also a GLMB with parameters [38]

$$\boldsymbol{\pi}_+ = \left\{\left(w_{Z_+}^{(I_+,c,\theta_+)},p_{Z_+}^{(c,\theta_+)}\right)\right\}_{(I_+,c,\theta_+)\in\mathcal{F}(\mathbb{L}_+)\times\mathbb{C}\times\Theta_+} \tag{5}$$

where

$$w_{Z_+}^{(I_+,c,\theta_+)} \propto \sum_I w^{(I,c)}\omega_{Z_+}^{(I,c,I_+,\theta_+)}, \tag{6}$$

$$\omega_{Z_+}^{(I,c,I_+,\theta_+)} = 1_{\Theta_+(I_+)}(\theta_+)\left[1-\bar{P}_S^{(c)}\right]^{I-I_+}\left[\bar{P}_S^{(c)}\right]^{I\cap I_+}, \tag{7}$$

$$\times\left[1-r_{B,+}\right]^{\mathbb{B}_+-I_+} r_{B,+}^{\mathbb{B}_+\cap I_+}\left[\bar{\psi}_{Z_+}^{(c,\theta_+)}\right]^{I_+}, \tag{8}$$

$$\bar{P}_S^{(c)}(\ell) = \left\langle p^{(c)}(\cdot,\ell),P_S(\cdot,\ell)\right\rangle, \tag{9}$$

$$\bar{\psi}_{Z_+}^{(c,\theta_+)}(\ell_+) = \left\langle \bar{p}_+^{(c)}(\cdot,\ell_+),\psi_{Z_+}^{(\theta_+(\ell_+))}(\cdot,\ell_+)\right\rangle, \tag{10}$$

$$\bar{p}_+^{(c)}(x_+,\ell_+) = 1_{\mathbb{L}}(\ell_+)\frac{\left\langle P_S(\cdot,\ell_+)f_+(x_+|\cdot,\ell_+),p^{(c)}(\cdot,\ell_+)\right\rangle}{\bar{P}_S^{(c)}(\ell_+)} + 1_{\mathbb{B}_+}(\ell_+)\,p_{B,+}(x_+,\ell_+), \tag{11}$$

$$p_{Z_+}^{(c,\theta_+)}(x_+,\ell_+) = \frac{\bar{p}_+^{(c)}(x_+,\ell_+)\,\psi_{Z_+}^{(\theta_+(\ell_+))}(x_+,\ell_+)}{\bar{\psi}_{Z_+}^{(c,\theta_+)}(\ell_+)}. \tag{12}$$

The recursive propagation of the current filtering density (3) to the next time is more compactly expressed by a *GLMB joint prediction and update operator* $\Omega:\mathcal{G}_{\mathbb{L}}\to\mathcal{G}_{\mathbb{L}_+}$ defined by $\Omega(\boldsymbol{\pi};\boldsymbol{f}_B^{(\mathbb{B}_+)},Z_+)=\boldsymbol{\pi}_+$ according to (5)–(12), where $\boldsymbol{f}_B^{(\mathbb{B}_+)}$ is the next birth density and Z_+ is the next measurement set.

III. Large-Scale GLMB Filtering

Due to practical limitations on computational resources, the original implementation of the GLMB filter proposed in [36], [37] cannot accommodate a very large number of objects simultaneously. The main computational bottleneck occurs in the measurement update, which involves processing each component of the predicted GLMB density using Murty's algorithm to find the K most significant components according to their weights. If there are N labels in a component and M measurements in total, then the complexity of the update is $\mathcal{O}(K(N+M)^3)$. Murty's algorithm can be replaced with Gibbs sampling to reduce the computational complexity of processing each component down to $\mathcal{O}(KN^2M)$ [38]. However, the quadratic complexity in the number of objects will still render this algorithm infeasible for

tracking a large number of objects simultaneously. Arguably, any feasible solution for large-scale multi-object tracking should have a maximum computational complexity of approximately $\mathcal{O}(KNM \cdot log(NM))$.

In many practical multi-object scenarios, the objects are not uniformly distributed across the state space, but often in separate groups. This structure can be exploited to improve computational efficiency of the multi-object tracker. Rather than representing the entire multi-object density as one "large" GLMB, we can approximate it as a product of much "smaller" GLMBs, herein referred to as *label-partitioned GLMB*s. This is based on the premise that a large GLMB, with well-separated groups of estimated objects, is decomposable into a product of almost independent smaller GLMBs. Consequently, such an approximation results in a negligible loss of information, whilst providing significant gains in computational efficiency.

A. Label-Partitioned GLMB

A labeled RFS density on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$ is said to be *label-partitioned* if it can be written as the following product

$$\boldsymbol{\pi}_{\mathfrak{L}}(\boldsymbol{X}) = \prod_{L \in \mathfrak{L}} \boldsymbol{\pi}_{\mathfrak{L}}^{(L)}(\boldsymbol{X} \cap (\mathbb{X} \times L)),$$

where $\mathfrak{L}$ is some partition of the label space $\mathbb{L}$, and each factor $\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}$ is a labeled RFS density on $\mathcal{F}(\mathbb{X} \times L)$. Note that for all $\boldsymbol{X} \in \mathcal{F}(\mathbb{X} \times \mathbb{L})$, $\boldsymbol{X} = \biguplus_{L \in \mathfrak{L}} \boldsymbol{X} \cap (\mathbb{X} \times L)$,[1] and hence $\{\mathcal{F}(\mathbb{X} \times L)\}_{L \in \mathfrak{L}}$ is also a partition of $\mathcal{F}(\mathbb{X} \times \mathbb{L})$. We call $\boldsymbol{\pi}_{\mathfrak{L}}$, denoted by its factors $\{\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}\}_{L \in \mathfrak{L}}$, an $\mathfrak{L}$-*partitioned* labeled RFS density. Further, if each factor $\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}$ is a GLMB, then $\boldsymbol{\pi}_{\mathfrak{L}}$ is said to be an $\mathfrak{L}$-partitioned GLMB on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$. We denote by $\mathcal{G}_{\mathbb{L}}(\mathfrak{L})$ the space of all $\mathfrak{L}$-*partitioned* GLMBs on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$.

Suppose that the current filtering density $\boldsymbol{\pi}_{\mathfrak{L}}$ is an $\mathfrak{L}$-partitioned GLMB on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$. Then, the prediction to the next time is also an $\mathfrak{L}$-partitioned GLMB. However, the resulting filtering density generally does not take on the same form. While the new filtering density $\boldsymbol{\pi}_{+}$ is still a GLMB that, in principle, can be computed [36]–[38], a direct computation via expansion of the product and subsequent update is not scalable in practice. Moreover, due to object births, deaths and transitions, the current partition $\mathfrak{L}$ of $\mathbb{L}$ is unsuitable as a basis for approximating the new filtering density.

Nonetheless, when estimated objects in the new filtering density also occur in separate groups, as do the sets of measurements which are associated with different groups of objects, it is possible to exploit this structure to improve computational efficiency. To maintain scalability and parallelisability, we approximate the new filtering density as a label-partitioned GLMB. This entails selection of the optimal partition of labels and measurements, and the optimal association of groups of labels to groups of measurements, in some statistical sense. Intuitively the selection of the partitions and associations should result in negligible statistical dependence between labels and measurements across different groups.

Let $\mathcal{P}(\mathbb{L})$, $\mathcal{P}(\mathbb{L}_+)$ and $\mathcal{P}(Z_+)$ denote the sets of all partitions of the current label space $\mathbb{L}$, the new label space $\mathbb{L}_+$ and the new measurements Z_+ respectively. Suppose that the current filtering density is an $\mathfrak{L}$-partitioned GLMB on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$,

$$\boldsymbol{\pi}_{\mathfrak{L}} = \left\{\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}\right\}_{L \in \mathfrak{L}}. \tag{13}$$

Consider a new partition $\mathfrak{L}_+ \in \mathcal{P}(\mathbb{L}_+)$, where each set of labels $L \in \mathfrak{L}_+$ is associated with a set of gated measurements $Z_+^{(L)} \subset Z_+$, such that $Z_+^{(I)} \cap Z_+^{(J)} = \emptyset$ for $I \neq J$. We seek to approximate the new filtering density as an $\mathfrak{L}_+$-partitioned GLMB on $\mathcal{F}(\mathbb{X} \times \mathbb{L}_+)$.

For a particular choice of $\mathfrak{L}_+ \in \mathcal{P}(\mathbb{L}_+)$ and $\mathfrak{Z}_+ \in \mathcal{P}(Z_+)$, let $\mathcal{A}_+(\mathfrak{L}_+, \mathfrak{Z}_+)$ denote the space of all positive 1-1 mappings $A_+ : \mathfrak{L}_+ \to \{\emptyset\} \uplus \mathfrak{Z}_+$, where positive 1-1 means 1-1 on $\mathfrak{Z}_+$. In other words, a given $A_+ \in \mathcal{A}_+(\mathfrak{L}_+, \mathfrak{Z}_+)$ maps each group of labels in the partition $\mathfrak{L}_+$, to either $\{\emptyset\}$ or a unique group of measurements in the partition $\mathfrak{Z}_+$, i.e. $A_+(L)$ is the set of measurements corresponding to the set of labels L. Ideally, we seek the approximation

$$\boldsymbol{\pi}_{\mathfrak{L}_+, \mathfrak{Z}_+, A_+} = \left\{\boldsymbol{\pi}_{\mathfrak{L}_+, \mathfrak{Z}_+, A_+}^{(L)}\right\}_{L \in \mathfrak{L}_+} \tag{14}$$

$$\boldsymbol{\pi}_{\mathfrak{L}_+, \mathfrak{Z}_+, A_+}^{(L)} = \Omega\left(\boldsymbol{\pi}_{\mathfrak{L}}; \boldsymbol{f}_B^{(L \cap \mathbb{B}_+)}, A_+(L)\right) \tag{15}$$

where

$$(\mathfrak{L}_+, \mathfrak{Z}_+, A_+) = \underset{\substack{\mathfrak{L}_+ \in \mathcal{P}(\mathbb{L}_+) \\ \mathfrak{Z}_+ \in \mathcal{P}(Z_+) \\ A_+ \in \mathcal{A}(\mathfrak{L}_+, \mathfrak{Z}_+)}}{\arg\min} D_{KL}\left(\boldsymbol{\pi}_+, \boldsymbol{\pi}_{\mathfrak{L}_+, \mathfrak{Z}_+, A_+}\right) \tag{16}$$

$$\text{s.t. } |L| \leq L_{\max}, \forall L \in \mathfrak{L}_+ \tag{17}$$

and $D_{KL}(\cdot, \cdot)$ denotes the Kullback-Leibler divergence (KLD). Here $L_{\max}$ is a user parameter determined by the available computational resources. Higher values of $L_{\max}$ result in a more accurate approximation to the multi-object filtering density, however, more memory and faster processing will be required to achieve real-time performance. Smaller values yield a coarser approximation, but real-time performance is achievable with less computational resources. Choosing $L_{\max} = 1$ is equivalent to running parallel Bernoulli filters [3, Section 14.7].

The combinatorial optimization problem in (16) is intractable, since the space of partitions is prohibitively large. We propose an approximate two-step solution which is tractable for large-scale problems. The first step involves choosing a suboptimal partition of $\mathbb{L}_+$, subject to a constraint on the maximum group cardinality. The second step involves parallel computation of the factors in the new filtering density, directly as a product of factors from the old filtering density. This strategy avoids the explicit expansion of the product, and subsequent refactorization after update, which would be intractable. These steps are described in the following two subsections.

B. Label Partitioning

We aim to find a partition of the label space such that any pair of labels that do not appear in the same group are approximately statistically independent. In the standard multi-object tracking model, as defined in Sections II-A and II-B, this statistical dependence arises solely via the uncertainty in the unknown association between measurements and objects. That is, if a particular detection could have originated from one of several objects, then there will be a statistical dependence between those objects in the multi-object filtering density.

[1]The disjoint union notation ($\uplus$), in expressions involving unions over a partition, would be equivalent to the standard union. However, we use it to serve as a reminder that the constituent sets of the union are disjoint.

Intuitively, tracks that are well-separated in the measurement space will have low statistical dependence, because the probability that these tracks give rise to closely spaced measurements is extremely low. In this context, "well-separated" means that the distance between tracks in the measurement space is large compared to the measurement noise and the uncertainty in the objects' predicted location. This is the key property that we exploit in order to partition the label space in a way that minimises the amount of potential measurement sharing between objects in different groups.

In principle, this can be achieved by analysing the distributions of the predicted measurements corresponding to all objects represented in a GLMB. Suppose each factor has the form

$$\boldsymbol{\pi}_{\mathfrak{L}}^{(L)} = \left\{\left(w_{\mathfrak{L},L}^{(I,c)}, p_{\mathfrak{L},L}^{(c)}\right)\right\}_{(I,c)\in\mathcal{F}(L)\times\mathbb{C}(L)} \tag{18}$$

Then for each label $\ell \in \mathbb{L}_+$ we compute the distribution of the predicted measurement

$$\tilde{p}_+(z,\ell) = \langle g(z|\cdot,\ell), p_+(\cdot,\ell)\rangle \tag{19}$$

via the distribution of the predicted state

$$\begin{aligned} p_+(x,\ell) \propto\ & 1_{\mathbb{B}_+}(\ell)\, r_B(\ell)\, p_B(x,\ell) \\ & + \sum_{L\in\mathfrak{L}} \sum_{\substack{(I^{(L)},c^{(L)})\in \\ \mathcal{F}(L)\times\mathbb{C}(L)}} 1_L(\ell)\, 1_{I^{(L)}}(\ell) \\ & \times w_{\mathfrak{L},L}^{(I^{(L)},c^{(L)})} \left\langle p_{\mathfrak{L},L}^{(c^{(L)})}(\cdot), f(x|\cdot,\ell)\right\rangle. \end{aligned} \tag{20}$$

The distribution $\tilde{p}_+(\cdot,\ell)$ can be used to construct a "measurement gating region," $B(\ell)\subseteq\mathbb{Z}$ for each label $\ell\in\mathbb{L}_+$, which contains the majority of the probability mass for the predicted measurement. These gating regions are the basis for partitioning the next label space $\mathbb{L}_+$. A partition $\mathfrak{L}_+ = \{L_1,\ldots,L_{|\mathfrak{L}|}\}$ is formed by splitting $\mathbb{L}_+$ into groups of labels whose corresponding gating regions intersect (either directly, or indirectly via a sequence of labels in the same group), and where there is no intersection between gating regions for labels in different groups. Furthermore, for computational feasibility, the gating regions must be sufficiently small such that $\mathbb{L}_+$ can be partitioned into groups no larger than some predefined cardinality threshold $L_{\max}$. That is, $\mathfrak{L}$ must satisfy the following three conditions:

1) For all $L\in\mathfrak{L}_+$, and for any $\ell_i,\ell_j\in L$, either $B(\ell_i)\cap B(\ell_j)\neq\emptyset$ or there exists $\{\ell_1,\ldots,\ell_n\}\subseteq L$ such that $B(\ell_i)\cap B(\ell_1)\neq\emptyset, B(\ell_1)\cap B(\ell_2)\neq\emptyset,\ldots,B(\ell_{n-1})\cap B(\ell_n)\neq\emptyset, B(\ell_n)\cap B(\ell_j)\neq\emptyset$
2) $[\bigcup_{\ell\in L_i} B(\ell)]\cap[\bigcup_{\ell\in L_j} B(\ell)]=\emptyset$, for all $i,j\in\{1,\ldots|\mathfrak{L}_+|\}, i\neq j$,
3) $|L|\leq L_{\max}$ for all $L\in\mathfrak{L}_+$.

Under these conditions, the label space is partitioned such that there is no overlap between the regions of $\mathbb{Z}$ corresponding to the groups of labels represented by $\mathfrak{L}_+$, taking into account the prediction and likelihood. Consequently the multi-object filtering density at the next time should exhibit negligible statistical dependence between different groups of the partition $\{\boldsymbol{X}_+^{(L)} = \boldsymbol{X}_+\cap\mathcal{F}(\mathbb{X}\times L) : L\in\mathfrak{L}_+\}$. Hence, we can assume that the multi-object likelihood can be well-approximated by the following separable form

$$\hat{g}\left(\biguplus_{L\in\mathfrak{L}_+} Z_+^{(L)} \middle| \biguplus_{L\in\mathfrak{L}_+} \boldsymbol{X}_+^{(L)}\right) \triangleq \prod_{L\in\mathfrak{L}_+} g\left(Z_+^{(L)}|\boldsymbol{X}_+^{(L)}\right) \tag{21}$$

which facilitates a fast parallel evaluation of the label-partitioned GLMB filtering density, as described in subsection III-C.

A naive approach to partitioning $\mathbb{L}_+$, subject to the conditions above, will have computational complexity $\mathcal{O}(|\mathbb{L}_+|^2)$ since all possible label pairs must be examined to determine whether their gating regions intersect. This is clearly infeasible for large-scale tracking problems. Fortunately, techniques in computational geometry can be applied to dramatically reduce the computational complexity of the partitioning, thereby making it feasible for such problems. A method for efficient implementation is discussed in Section IV-A.

C. Computing Label-Partitioned GLMB Filtering Density

Suppose the current filtering density is an $\mathfrak{L}$-partitioned GLMB, and that we are given a new partition $\mathfrak{L}_+$ of $\mathbb{L}_+$. The goal is to approximate the filtering density at the next time as an $\mathfrak{L}_+$-partitioned GLMB. To achieve this in a way that is scalable to problems involving a very large number of objects, we consider approximating the current filtering density according to a modified partition structure $\mathfrak{S} = \{L\cap\mathbb{L} : L\in\mathfrak{L}_+\}$, i.e. we take each element of the new partition $\mathfrak{L}_+$, and intersect it with the current label set $\mathbb{L}$. The current filtering density can then be approximated as an $\mathfrak{S}$-partitioned GLMB

$$\boldsymbol{\pi}_{\mathfrak{S}} = \left\{\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}\right\}_{S\in\mathfrak{S}} \tag{22}$$

that mimimises $D_{KL}(\boldsymbol{\pi}_{\mathfrak{L}};\boldsymbol{\pi}_{\mathfrak{S}})$. This approximation is given explicitly in Proposition 1 (see Appendix A for proof).

Proposition 1: Given an $\mathfrak{L}$-partitioned GLMB $\boldsymbol{\pi}_{\mathfrak{L}} = \{\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}\}_{L\in\mathfrak{L}}$ on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$, and suppose that $\mathfrak{S}$ is another partition of $\mathbb{L}$. Then the $\mathfrak{S}$-partitioned labeled RFS density $\boldsymbol{\pi}_{\mathfrak{S}} = \{\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}\}_{S\in\mathfrak{S}}$ that minimises $D_{KL}(\boldsymbol{\pi}_{\mathfrak{L}};\boldsymbol{\pi}_{\mathfrak{S}})$, is an $\mathfrak{S}$-partitioned GLMB, with GLMB factors

$$\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}\left(\boldsymbol{X}^{(S)}\right) = \prod_{L\in\mathfrak{L}} \boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}\left(\boldsymbol{X}^{(S)}\right)$$

where

$$\boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)} = \left\{\left(w_{\mathfrak{L},\mathfrak{S},L,S}^{(H,c)}, p_{\mathfrak{L},\mathfrak{S},L,S}^{(c)}\right)\right\}_{(H,c)\in\mathcal{F}(L\cap S)\times\mathbb{C}(L)}$$

$$w_{\mathfrak{L},\mathfrak{S},L,S}^{(H,c)} = \sum_{W\in\mathcal{F}(L-S)} w_{\mathfrak{L},L}^{(H\cup W,c)}$$

$$p_{\mathfrak{L},\mathfrak{S},L,S}^{(c)}(x,\ell) = 1(\ell)p_{\mathfrak{L},L}^{(c)}(x,\ell)$$

for each $(L,S)\in\mathfrak{L}\times\mathfrak{S}$ such that $L\cap S\neq\emptyset$.

Given the approximation (22) to the current filtering density, and the separable multi-object likelihood (21), the joint prediction and update can be applied to all factors of $\boldsymbol{\pi}_{\mathfrak{S}_+}$ in parallel, yielding the new $\mathfrak{L}_+$-partitioned GLMB filtering density,

$$\boldsymbol{\pi}_{\mathfrak{L}_+,+} = \left\{\boldsymbol{\pi}_{\mathfrak{L}_+,+}^{(J)}\right\}_{J\in\mathfrak{L}_+}. \tag{23}$$

An explicit expression for (23) is given in Proposition 2 (see Appendix A for proof). Further details for efficient implementation are discussed in Sections IV-B and IV-C.

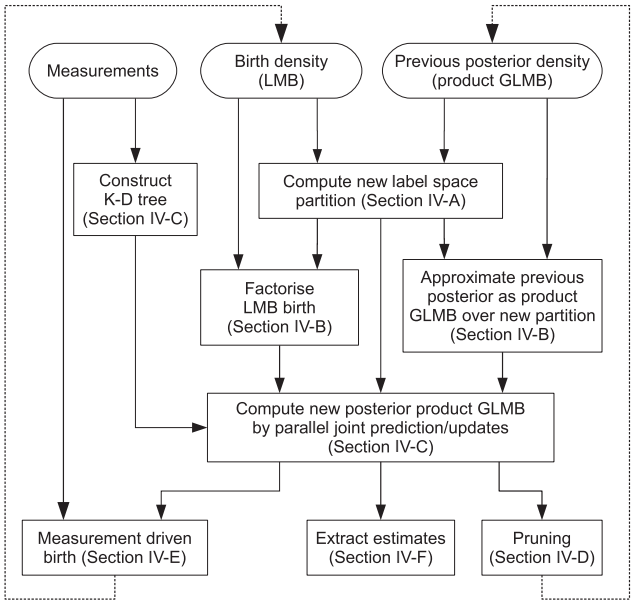


Fig. 1. High-level flow diagram of the large-scale GLMB filter.

Proposition 2: Given a separable multi-object likelihood (21), where $\mathfrak{L}_+$ is partition of $\mathbb{L}_+$, and suppose that the current multi-object filtering density is an $\mathfrak{S}$ -partitioned GLMB of the form $\boldsymbol{\pi}_{\mathfrak{S}} = \{\boldsymbol{\pi}_{\mathfrak{S}}^{(J\cap\mathbb{L})}\}_{J\in\mathfrak{L}_+}$ where

$$\boldsymbol{\pi}_{\mathfrak{S}}^{(J\cap\mathbb{L})} = \left\{\left(w_{\mathfrak{S},J\cap\mathbb{L}}^{(I,c)}, p_{\mathfrak{S},J\cap\mathbb{L}}^{(c)}\right)\right\}_{(I,c)\in\mathcal{F}(J\cap\mathbb{L})\times\mathbb{C}(J)}.$$

Then the multi-object filtering density at the next time is the $\mathfrak{L}_+$-partitioned GLMB $\boldsymbol{\pi}_{\mathfrak{L}_+,+} = \{\boldsymbol{\pi}_{\mathfrak{L}_+,+}^{(J)}\}_{J\in\mathfrak{L}_+}$ where

$$\boldsymbol{\pi}_{\mathfrak{L}_+,+}^{(J)} = \Omega\left(\boldsymbol{\pi}_{\mathfrak{S}}^{(J\cap\mathbb{L})}; \boldsymbol{f}_B^{(J\cap\mathbb{B}_+)}, Z_+^{(J)}\right)$$

and $\Omega(\cdot; \boldsymbol{f}_B^{(\mathbb{B}_+)}, Z_+) : \mathcal{G}_{\mathbb{L}} \to \mathcal{G}_{\mathbb{L}_+}$ is the joint prediction and update operator.

IV. Implementation

In this section we describe in more detail our implementation of a large-scale GLMB filter, based on the concepts introduced in the previous section. The algorithm is composed of several modules as shown in Fig. 1. The details of each of these modules are discussed in the following subsections.

A. Step 1: Label Space Partition Selection

We proceed under the standard assumptions of linear-Gaussian transition and measurement models

$$f_+(x_+|x,\ell) = \mathcal{N}(x_+; Fx, Q),$$
$$g(z|x,\ell) = \mathcal{N}(z; Hx, R).$$

If the prior distribution for track ℓ is a Gaussian mixture

$$p(x;\ell) = \sum_{u=1}^{U(\ell)} w^{(u)}(\ell)\mathcal{N}\left(x, m^{(u)}(\ell), P^{(u)}(\ell)\right)$$

then the distribution of the predicted measurement is also a Gaussian mixture

$$\tilde{p}_+(z;\ell) = \sum_{u=1}^{U(\ell)} w^{(u)}(\ell)\mathcal{N}\left(z, \tilde{m}^{(u)}(\ell), \tilde{P}^{(u)}(\ell)\right), \quad (24)$$

$$\tilde{m}^{(u)}(\ell) = HFm^{(u)}(\ell), \quad (25)$$

$$\tilde{P}^{(u)}(\ell) = R + H(Q + FPF^T)H^T. \quad (26)$$

For tractability, the distribution of the predicted measurement is then approximated as a uniform mixture

$$\tilde{p}_+(z,\ell) \approx \sum_{u=1}^{U(\ell)} \mathcal{U}\left(z; B^{(u)}(\ell)\right) \quad (27)$$

where $\mathcal{U}(\cdot; B^{(u)}(\ell))$ is a uniform distribution on an axis-aligned hyper-rectangle $B^{(u)}(\ell)$ that should correspond to the region where $\mathcal{N}(\cdot, \tilde{m}^{(u)}(\ell), \tilde{P}^{(u)}(\ell))$ has significant mass. An efficient method for computing this is to consider the ellipsoidal gate centered on $\tilde{m}^{(u)}(\ell)$ and shaped according to $\tilde{P}^{(u)}(\ell)$, which has probability P_G of containing the received measurement. The axis-aligned hyper-rectangle $B^{(u)}(\ell)$ can be chosen as a tight bounding box on the ellipsoidal gate, which can be computed in simple closed form as a function of $\tilde{m}^{(u)}(\ell)$, $\tilde{P}^{(u)}(\ell)$ and P_G. For any $L \subseteq \mathbb{L}_+$, define the prediction gate

$$B(L) = \bigcup_{\ell\in L}\left[\bigcup_{u=1}^{U(\ell)} B^{(u)}(\ell)\right]. \quad (28)$$

We seek a partition $\mathfrak{L}_+$ of $\mathbb{L}_+$ which maximises $|\mathfrak{L}_+|$ and satisfies

$$B(I)\cap B(J) = \emptyset, \forall I, J \in \mathfrak{L}_+, I \neq J. \quad (29)$$

A solution can always be found via the following procedure with $\mathcal{O}(R_T \log_d R_T + R_I + R_L)$ complexity, where d is the dimension of the measurement space, R_T is the total number of hyper-rectangles, R_I is the number of intersecting hyper-rectangle pairs, and R_L is the total number of labels at the current time. A segment-tree [44]–[46] is first constructed containing all hyper-rectangles, which is used to find all intersecting prediction gates, with complexity $O(R_T \log R_T + R_I)$. A graph is then constructed, consisting of one node for each label at the current time, and an edge for every pair of labels whose prediction gates intersect. The connected components of the graph are found via depth-first search, which has $\mathcal{O}(R_L)$ complexity, and can be further accelerated via parallel processing. The connected components then give the desired partition of $\mathbb{L}_+$. However, if the additional constraint $|L| \leq L_{\max}, \forall L \in \mathfrak{L}_+$ is not satisfied, it is necessary to reduce the value for P_G and repeat the procedure until a feasible solution is found.

B. Step 2: Birth Factorisation and GLMB Repartitioning

Once an appropriate label space partition $\mathfrak{L}_+$of $\mathbb{L}_+$ has been found, we proceed to factorise the LMB birth density $\boldsymbol{f}_B^{(\mathbb{B}_+)}$ as a product over the partition $\mathfrak{B}_+ = \{L\cap\mathbb{B}_+ : L \in \mathfrak{L}_+\}$ of $\mathbb{B}_+$, and to approximate the current filtering density $\pi_{\mathfrak{L}}$ as a product over the partition $\mathfrak{S} = \{L\cap\mathbb{L} : L \in \mathfrak{L}_+\}$ of $\mathbb{L}$.

The LMB birth is exactly factorised as a $\mathfrak{B}_+$-partitioned LMB $\boldsymbol{f}_B^{(\mathbb{B}_+)} = \{\boldsymbol{f}_B^{(B_+)}\}_{B_+\in\mathfrak{B}_+}$. The minimum KLD approximation

to $\boldsymbol{\pi}_{\mathfrak{L}}$ as an $\mathfrak{S}$-partitioned GLMB, $\boldsymbol{\pi}_{\mathfrak{S}} = \{\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}\}_{S\in\mathfrak{S}}$, is obtained via Proposition 1. Consider a given $S \in \mathfrak{S}$, then for each $L \in \mathfrak{L}$, we calculate the GLMB $\boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}$ for labels which are common to both S and L. The expression for $\boldsymbol{\pi}_{\mathfrak{S}}^{(S)} = \prod_{L\in\mathfrak{L}} \boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}$ is given by expanding the product over $L \in \mathfrak{L}$. In practice, this does not necessary require enumerating all pairs (L, S), because some combinations may satisfy $L \cap S = \emptyset$ resulting in a trivial expression for $\boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}$. It is possible to identify pairs for which $L \cap S \neq \emptyset$ by querying an appropriate map structure used to represent the GLMB factors with logarithmic complexity. In addition, the computation of the final product for $\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}$ can be efficiently implemented using a k-shortest path algorithm or stochastic sampling strategy to truncate the result without explicit expansion. Note that this factorisation and repartitioning step is trivially parallelisable in each of the factors.

C. Step 3: Parallel Propagation of Label-Partitioned GLMB

Given the new partition $\mathfrak{L}_+$ of $\mathbb{L}_+$, in preparation for the update, it is necessary to compute for each $L \in \mathfrak{L}_+$, the non intersecting measurement sets $Z_+^{(L)} = Z_+ \cap B(L)$ that fall inside the region $B(L)$ found in Step 1. This will automatically satisfy $Z_+^{(I)} \cap Z_+^{(J)} = \emptyset$ since $B(I) \cap B(J) = \emptyset$ for any $I, J \in \mathfrak{L}_+$. A K-dimensional (K-D) tree data structure [47], can be used to find $Z_+^{(L)}$ efficiently with $\mathcal{O}(R_U|Z_+|^{1-1/d} + R_Z)$ complexity, where d is the dimension of the measurement space, $R_U = \sum_{\ell\in L} U(\ell)$ and $R_Z = \sum_{\ell\in L} |Z_+ \cap B(\ell)|$.

Since both the likelihood and prediction can be written as products over the same partition, the factors of the $\mathfrak{L}_+$-partitioned GLMB filtering density can be computed in parallel independently as shown in Proposition 2. This is a key feature that allows us to address large-scale multi-target tracking problems, by utilising modern multi-core architectures. For each $L \in \mathfrak{L}_+$, computation of the filtering density is given by Proposition 2, via the GLMB joint prediction and update operator Ω with inputs $\boldsymbol{\pi}_{\mathfrak{S}}^{(L\cap\mathbb{L})}$, $\boldsymbol{f}_B^{(L\cap\mathbb{B}_+)}$, $Z_+^{(L)}$. Our implementation follows [38], making use of a Gibbs sampler to efficiently generate GLMB components with high filtering weights, while also maintaining diversity across the generated samples.

Finally, an "association probability" is evaluated for each measurement in Z_+, which is used to generate the LMB birth density for the subsequent iteration of the filter. For a measurement $z \in Z_+$ that falls inside the bounding region $B(L)$, this probability can be computed after the update of the factor corresponding to L, by summing the weights of all GLMB components in which the measurement was assigned to an object. For measurements that do not fall inside any bounding region, the corresponding association probability is set to zero.

D. Step 4: Pruning Label-Partitioned GLMB Filtering Density

To improve computational efficiency, for each factor of the GLMB filtering density we prune its constituent components by removing those whose contributions are deemed to be insignificant. In previous implementations of the GLMB filter [36]–[38], a simple component pruning procedure was used, whereby a fixed number of highest weighted components were retained after each iteration, or components with weights below a certain threshold were deleted.

In addition, pruning is necessary for entire factors of the GLMB filtering density, also with the aim of improving the algorithm's computational efficiency. The idea is to eliminate entire factors which have negligible contribution to the new filtering density. A simple criterion is the probability that a factor has zero cardinality. Factors for which this probability exceeds a certain threshold are simply deleted. An alternative is to retain only a fixed number of factors with the highest probability of having non-zero cardinality.

E. Step 5: Measurement Driven Birth

The association probabilities computed in step 2 capture the likelihood that the given measurement originated from *any* one of the existing objects. These probabilities are now used to construct a labelled multi-Bernoulli distribution, which will serve as the birth density for the next iteration of the filter. Each measurement with association probability below some pre-defined threshold is used to generate a component of the LMB distribution. The measurement itself, along with prior distributions on the unobserved state components, are used to generate a birth density for each component. This so-called "measurement-driven" approach requires fewer prior assumptions regarding the initial state of newborn objects than static birth models.

F. Step 6: Estimate Extraction

A parallelisable strategy for extracting labelled estimates of the current object states is to process each factor independently using standard approaches. The estimates for a particular factor can be obtained by first finding the maximum a-posteriori (MAP) cardinality estimate for the number of objects, then finding the highest weighted component with the relevant cardinality, and finally selecting either the MAP/EAP estimates for each label. The overall multi-object estimate is obtained by taking the union of the estimates from all factors. These are subsequently used to update the track estimates, by matching up the estimates with corresponding labels across different times.

V. Multi-object Tracking Performance Metric

In [41], the optimal sub-pattern assignment (OSPA) metric was proposed as a mathematically consistent and physically meaningful distance between two sets of points. This has found widespread use in the evaluation of multi-object tracking performance, where it has become a common practice to present a plot of the OSPA distance between the estimated and true multi-object states versus time. While this can provide an indication of the multi-object tracking performance, it does not fully account for errors between the estimated and true sets of tracks. Specifically, the OSPA distance between multi-object states does not penalise phenomena such as track switching and fragmentation in a consistent fashion.

In [39], a metric called OSPA for tracks (OSPA-T) was proposed, along with a method for its approximate computation. The disadvantage of this approach is that the approximation no longer satisfies the axioms of a metric, and thus it may not behave as one would expect. A number of drawbacks of the approximate OSPA-T were discussed in [40], in which the authors defined another metric that alleviates these drawbacks. However, this metric is computationally intractable for most practical problems. In [48], two metrics inspired by the CLEAR

MOT heuristics for tracking performance evaluation in computer vision were proposed. The drawbacks of these metrics were pointed out in [49], and two other metrics were proposed. However, these metrics are not suitable for evaluating multi-object tracking performance in large-scale scenarios.

Since the OSPA distance between two sets is constructed from the base-distance between the elements of the sets, a simple way of using it to evaluate multi-object tracking performance is to choose the base-distance to be a distance between tracks. When this base-distance is also constructed via OSPA, the result is the OSPA metric on an OSPA base-distance, which we call OSPA$^{(2)}$ to distinguish it from the standard use. A meaningful OSPA$^{(2)}$ depends on a meaningful base-distance between tracks.

This section presents the OSPA$^{(2)}$ metric together with a physically meaningful base-distance between tracks. We also develop a scalable algorithm for computing this OSPA$^{(2)}$ distance (exactly), capable of evaluating large-scale tracking performance. We begin by defining the following notation:

- $\mathbb{T} = \{1, 2, \ldots, K\}$ is a finite space of time indices (representing times $\{t_1, t_2, \ldots, t_K\}$), which includes all time indices from the beginning to the end of the scenario.
- $\mathbb{X}$ is the single-object state space, and $\mathcal{F}(\mathbb{X})$ is the space of finite subsets of $\mathbb{X}$.
- $\mathbb{U}$ is the space of all functions mapping time indices in $\mathbb{T}$ to state vectors in $\mathbb{X}$, i.e. $\mathbb{U} = \{f : \mathbb{T} \mapsto \mathbb{X}\}$. We refer to each element of $\mathbb{U}$ as a *track*.
- For any $f \in \mathbb{U}$, its domain, denoted by $\mathcal{D}_f \subseteq \mathbb{T}$, is the set of time instants at which the object exists.

Also, recall that for a function $d(\cdot, \cdot)$ to be called a "metric," it must satisfy the following four properties:

P1. $d(x, y) \geq 0$ (non-negativity),
P2. $d(x, y) = 0 \iff x = y$ (identity),
P3. $d(x, y) = d(y, x)$ (symmetry),
P4. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

The OSPA distance $d_p^{(c)}(\phi, \psi)$ between $\phi, \psi \in \mathcal{F}(\mathbb{X})$ with order p and cutoff c is defined as follows [41]. For $\phi = \{\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(m)}\}$ and $\psi = \{\psi^{(1)}, \psi^{(2)}, \ldots, \psi^{(n)}\}$, $m \leq n$

$$d_p^{(c)}(\phi, \psi) \triangleq \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} \bar{d}^{(c)}\left(\phi^{(i)}, \psi^{(\pi(i))}\right)^p + c^p (n - m) \right) \right)^{1/p} \tag{30}$$

where $\bar{d}^{(c)}(\phi^{(i)}, \psi^{(i)}) \triangleq \min(c, d(\phi^{(i)}, \psi^{(i)}))$, in which $d(\cdot, \cdot)$ is a metric on the space $\mathbb{X}$. If $m > n$, then $d_p^{(c)}(\phi, \psi) \triangleq d_p^{(c)}(\psi, \phi)$. Additionally, $d_p^{(c)}(\emptyset, \phi) \triangleq c$, and $d_p^{(c)}(\emptyset, \emptyset) \triangleq 0$.

Note that in (30), the factor of $1/n$, which normalises the distance by the number of objects, is crucial for the OSPA to have the intuitive interpretation as a per-object error.

A. Base-Distance Between Tracks

We now use the OSPA distance (30) to construct a base-distance as a metric on the space $\mathbb{U}$ of tracks. One simple base-distance is [42]:

$$\tilde{d}^{(c)}(x, y) = \frac{1}{T} \sum_{t=1}^{T} d^{(c)}\left(\{x(t)\}, \{y(t)\}\right) \tag{31}$$

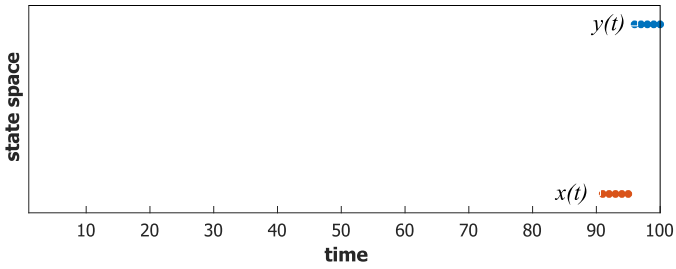


Fig. 2. Hypothetical scenario with two well-separated tracks inside a relatively long time window.

where

$$d^{(c)}(\phi, \psi) = \begin{cases} 0, & |\phi| = |\psi| = 0 \\ c, & |\phi| \neq |\psi| \\ \min\left(c, d\left(\phi^{(1)}, \psi^{(1)}\right)\right), & |\phi| = |\psi| = 1 \end{cases}. \tag{32}$$

Note that in (31) $\{x(t)\}$ is a singleton if $t \in \mathcal{D}_x$, or $\{x(t)\}$ is empty if $t \notin \mathcal{D}_x$ (and likewise for $\{y(t)\}$). In this case, the OSPA distance defined in (30) reduces to (32). Hence, the order parameter p becomes redundant, and is omitted.

The base-distance (31) exhibits counter-intuitive behaviour [43], wherein a pair of well-separated tracks may have a smaller distance than expected. If there are two short tracks inside a much longer window, then averaging over this window may result in a relatively small distance, even if the tracks are separated by a distance of c. For example, consider the scenario illustrated in Fig. 2. Inside the window of length $T = 100$, there are exactly two tracks, x with domain $\mathcal{D}_x = \{91 : 95\}$, and y with domain $\mathcal{D}_y = \{96 : 100\}$. If the base-distance is calculated according to (31), i.e. as an average over $t \in \{1, \ldots, T\}$ then $\tilde{d}^{(c)}(x, y) = c/10$. Furthermore, as the length of the window increases, the value of this base-distance decreases, which is counter intuitive, as the two tracks do not overlap in time, and are indeed very far apart. Thus it would actually be expected that the base-distance assign the maximum penalty c.

This issue can be resolved by constructing the distance $\tilde{d}^{(c)}(x, y)$ between two tracks $x, y \in \mathbb{U}$ as the mean OSPA distance between the set of states defined by x and y, over all times $t \in \mathcal{D}_x \cup \mathcal{D}_y$, i.e.

$$\tilde{d}^{(c)}(x, y) = \begin{cases} \sum_{t \in \mathcal{D}_x \cup \mathcal{D}_y} \frac{d^{(c)}(\{x(t)\}, \{y(t)\})}{|\mathcal{D}_x \cup \mathcal{D}_y|}, & \mathcal{D}_x \cup \mathcal{D}_y \neq \emptyset \\ 0, & \mathcal{D}_x \cup \mathcal{D}_y = \emptyset \end{cases}. \tag{33}$$

Averaging over $\mathcal{D}_x \cup \mathcal{D}_y$, instead of $\{1, \ldots, T\}$ as per (31), results a base-distance with intuitively consistent behaviour. For the example in Fig. 2, this choice of base-distance gives $\tilde{d}^{(c)}(x, y) = c$. Notice that even if the window is expanded, the distance still evaluates to the cutoff value c, as we would expect. In order to use (33) as a base-distance between tracks, we need to establish that it defines a metric on $\mathbb{U}$. That is, it must satisfy the properties P1-P4 as previously mentioned.

Proposition 3: Let $d^{(c)}(\cdot, \cdot)$ be a metric on the finite subsets of $\mathbb{X}$, such that the distance between a singleton and an empty set assumes the maximum attainable value c. Then the distance between two tracks as defined by (33) is also a metric.

The proof of this proposition involves some rather lengthy algebraic manipulations, and as such, it has been relegated to Appendix B. This result establishes that (33) is indeed a metric on the space $\mathbb{U}$, when $d^{(c)}(\cdot,\cdot)$ is defined according to (32).

Before proceeding to define OSPA$^{(2)}$, we make two important observations regarding the properties of this base-distance.

- Since $d^{(c)}(\cdot,\cdot) \leq c$, the distance between tracks saturates at the value c, i.e. $\tilde{d}^{(c)}(\cdot,\cdot) \leq c$.
- For two tracks x and y such that $\mathcal{D}_x = \mathcal{D}_y$, (33) can be interpreted as a mean square error (MSE) between x and y. Hence, this base-distance can be regarded as a generalisation of the MSE for tracks of different domains.

B. OSPA$^{(2)}$ for Tracks

The distance between two tracks as defined in Section V-A is both a metric on the space $\mathbb{U}$, and bounded by the value c. It is therefore suitable to serve as a base-distance for the OSPA metric on the space of finite sets of tracks $\mathcal{F}(\mathbb{U})$. Let $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\} \subseteq \mathcal{F}(\mathbb{U})$ and $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\} \subseteq \mathcal{F}(\mathbb{U})$ be two sets of tracks, where $m \leq n$. We define the distance $\check{d}_p^{(c)}(X,Y)$ between X and Y as the OSPA with base-distance $\tilde{d}^{(c)}(\cdot,\cdot)$ (the time averaged OSPA given by equation (33)). That is,

$$\check{d}_p^{(c)}(X,Y) \triangleq \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}\tilde{d}^{(c)}\left(x^{(i)},y^{(\pi(i))}\right)^p + c^p\left(n-m\right)\right)\right)^{1/p}, \tag{34}$$

where c is the cutoff and p is the order parameter. If $m > n$, then $\check{d}_p^{(c)}(X,Y) \triangleq \check{d}_p^{(c)}(Y,X)$. Note also that $\check{d}_p^{(c)}(\emptyset,X) \triangleq c$, and $\check{d}_p^{(c)}(\emptyset,\emptyset) \triangleq 0$. We call this distance OSPA-on-OSPA or OSPA$^{(2)}$, which can be interpreted as the time-averaged per-track error.

1) Efficient Evaluation of OSPA$^{(2)}$: Evaluating (34) involves the following three steps:

1) Compute an $m \times n$ cost matrix C, with entries $C_{i,j} = \tilde{d}^{(c)}(x^{(i)}, y^{(j)})$, according to (33).
2) Solve a 2-D assignment problem with cost matrix C, to find the minimum cost 1-1 assignment of columns to rows.
3) Use the result of step 2 to evaluate $\check{d}_p^{(c)}(X,Y)$ via (34).

A basic implementation of this procedure would require computing the base-distance between all pairs of tracks in X and Y, which has complexity $\mathcal{O}(|w_k|mn)$. Step 2 then requires solving a dense optimal assignment problem with complexity $\mathcal{O}((m+n)^3)$. This would preclude its use in large-scale tracking scenarios involving millions of objects, as the cost matrix would consume too much memory, and the assignment problem would be infeasible.

Fortunately, in many practical applications, the base-distance between most pairs of tracks will saturate at the cutoff value c. This can be exploited to dramatically reduce the computational complexity, making it feasible for large-scale problems. Firstly, recall that the time averaging in the base-distance (33) is carried out only over the union of the track domains. Consequently, the base-distance between any pair of tracks that have no corresponding states closer than a distance of c, must saturate at c. Such pairs can be considered unassignable, and efficient spatial searching algorithms can be applied to extract only the assignable pairs of tracks in $\mathcal{O}(|w_k|m\log n)$ time. Once these have been found, a sparse optimal assignment algorithm can be used to obtain the lowest-cost matching between the ground truth and estimated tracks. Such algorithms can solve sparse assignment problems with a much lower average complexity than is possible under the dense optimal assignment formulation.

2) Visualisation of OSPA$^{(2)}$: In practice, it is desirable to examine the tracking performance as a function of time, so that trends in algorithm behaviour can be analysed in response to changing scenario conditions. This can be achieved by plotting

$$\alpha_k\left(X,Y;w_k\right) = \check{d}_p^{(c)}\left(X_{w_k},Y_{w_k}\right) \tag{35}$$

as a function of k, where w_k is a set of time indices that varies with k, and

$$X_{w_k} = \{x|_{w_k} : x \in X \text{ and } \mathcal{D}_x \cap w_k \neq \emptyset\}, \tag{36}$$

$$Y_{w_k} = \{y|_{w_k} : y \in Y \text{ and } \mathcal{D}_y \cap w_k \neq \emptyset\}, \tag{37}$$

where $f|_w$ denotes the restriction of f to domain w.

Note that the sets X_{w_k} and Y_{w_k} only contain those tracks whose domain overlaps with w_k, i.e. any tracks whose domain lies completely outside the set w_k are disregarded. Choosing different values for the set w_k allows us to examine the performance of tracking algorithms over different time scales. For example, a straightforward approach is to set $w_k = \{k-N+1, k-N+2, \ldots, k\}$, so that at time k, the set w_k consists of only the latest N time steps. In this case, choosing a small value for N will indicate the tracking performance over shorter time periods, while larger values will reveal the longer-term tracking performance. This choice is highly dependent on the application at hand. For example, in real-time surveillance, we may only be interested in tracking objects over a period of a few minutes, as older information may be considered irrelevant. In this case, a small value for N would suffice to capture the important aspects of the tracking performance. However, in an off-line scenario analysis application, we might require accurate trajectory estimates over much longer time periods, in which case a larger value for N would be more appropriate.

Furthermore, examining the same scenario using OSPA$^{(2)}$ with different window lengths could reveal important insights into the relationship between long-term and short-term performance. For example, the design of a multi-object tracking system often involves trade-offs between estimation accuracy and response time, and comparing OSPA$^{(2)}$ with long and short windows could help to characterise the nature of this trade-off.

Note that computing the OSPA$^{(2)}$ on a sliding window as described above, converges to the traditional OSPA (for sets of points) as N becomes smaller. For $N = 1$ the OSPA$^{(2)}$ becomes identical to the traditional OSPA.

It is important to understand that the OSPA$^{(2)}$ distance has a different interpretation to that of the traditional OSPA distance. Whereas the traditional OSPA distance captures the error between the true and estimated multi-target states at a single instant in time, the OSPA$^{(2)}$ distance captures the error between the true and estimated sets of tracks over a range of time instants, as determined by the choice of w_k. Therefore, careful consideration must be given to the design of w_k, and the user must be mindful of this when interpreting the results.

3) Behavior of the OSPA$^{(2)}$: The OSPA$^{(2)}$ metric exhibits intuitively consistent behavior when faced with various types of common tracking errors. For example, errors in localisation, cardinality, track fragmentation and track label switching, all yield expected increases in the OSPA$^{(2)}$ distance (illustrations

are shown in [42], [43]). Some of the key points that distinguish the OSPA$^{(2)}$ from the traditional OSPA distance are as follows:

- A track that is dropped and later regained with the same identity, yields a smaller increase in the OSPA$^{(2)}$ than a track that it dropped and regained with a different identity.
- When visualised according to the method described in the previous section, a longer window effectively sustains the influence of cardinality and track labelling errors for a longer duration, i.e. the metric "remembers" mistakes that were committed by the tracker further into the past.
- A greater frequency of track fragmentation and labelling errors results in a higher OSPA$^{(2)}$ distance.

Illustrations of how the OSPA$^{(2)}$ metric respond to specific scenario events can be found in [42], [43]. Also included is a larger synthetic example, in which the frequency of track fragmentation and label switching is varied in order to observe its influence on the OSPA$^{(2)}$ distance.

VI. Numerical Results

In this section, we demonstrate the proposed GLMB filter implementation on a simulated large-scale multi-object tracking scenario in which the peak number of objects appearing simultaneously exceeds one million. The scenario runs for 1000 time steps, on a rectangular 64 km by 36 km surveillance region. The single-object state consists of 2D position and velocity, i.e. state vectors have the form $[\,x\ \dot{x}\ y\ \dot{y}\,]^T$, with units of metres and metres per second.

New objects are generated throughout the scenario by sampling from an LMB distribution at each time step. This distribution consists of 20,000 components, where the density of the i-th component is a Gaussian $\mathcal{N}(\cdot; m^{(i)}, P)$ with $m^{(i)} = [\,x^{(i)}\ 0\ y^{(i)}\ 0\,]^T$ and $P = \mathrm{diag}[50, 5] \otimes I_2$, where the positional elements of $m^{(i)}$ are sampled according to $x^{(i)} \sim \mathcal{U}(0, 64000)$ and $y^{(i)} \sim \mathcal{U}(0, 36000)$. At a given time step k, the existence probabilities of all components in the birth model are set to a common value $r_{B,+}$, but different values for $r_{B,+}$ are used within different time intervals according to

$$r_{B,+} = \begin{cases} 0.15, & k \in [1, 400] \cup [501, 700] \\ 0.01, & k \in [401, 500] \cup [701, 1000] \end{cases},$$

which simulates a variable rate of object birth. With 20,000 birth components, the value $r_{B,+} = 0.15$ equates to an average birth rate of 3000 objects per time step, and $r_{B,+} = 0.01$ gives an average birth rate of 200 objects per time step. It is imperative to note that the filter is not provided with any information about the locations of the birth components nor their probabilities. Instead, it uses a measurement-based approach to adaptively construct an LMB birth distribution after each iteration.

At each time step, objects that existed at the previous time survive with probability $P_S = 0.999$, i.e. one out of every 1000 objects spontaneously dies on average. This survival probability is known by the filter. An object that has state x_k at time k and survives to time $k+1$, takes on a new state according to the discrete white noise acceleration model

$$x_{k+1} = Fx_k + Gw_k,$$
$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2, \quad G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \otimes I_2,$$

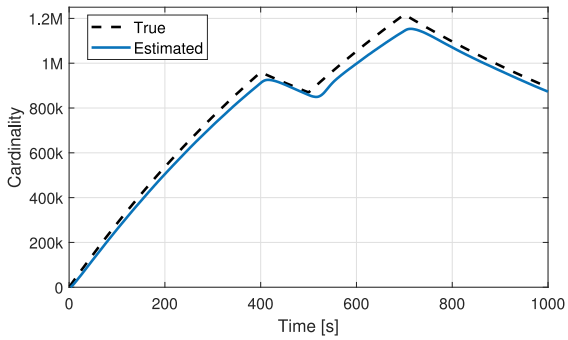


Fig. 3. True and estimated cardinality for large-scale tracking scenario.

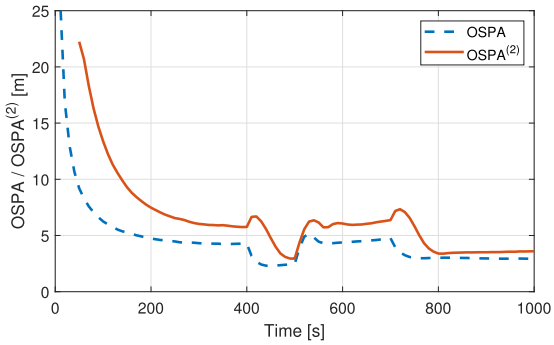


Fig. 4. OSPA and OSPA$^{(2)}$ distance for large-scale tracking scenario.

where $\otimes$ denotes the Kronecker product, and $w_k \sim \mathcal{N}(0, \sigma_w^2 I_2)$ is a 2×1 independent and identically distributed (i.i.d.) Gaussian process noise vector. In the current scenario, the sampling interval is $T = 1s$, and the process noise standard deviation is $\sigma_w = 0.2m/s^2$. The peak cardinality of approximately 1.2 million objects occurs at time 700, when the mean object density is around 520 per km^2.

We simulate data from a position sensor, corrupted by noise, missed detections and false alarms. An object that exists at time k with state x_k is detected with probability $P_D = 0.88$, in which case the object generates a measurement

$$z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k + v_k,$$

and $v_k \sim \mathcal{N}(0, \sigma_v^2 I_2)$ is a 2×1 i.i.d. Gaussian measurement noise vector, with $\sigma_v = 5m$. Each set of measurements also contains false alarms, the number of which is Poisson distributed with a mean of 200 per km^2 (i.e. an overall mean of $200 \times 64 \times 36 = 460,800$ per scan), and a spatial distribution that is uniform across the surveillance region.

For the filter, the number of GLMB components generated during the update of the i-th factor in the filtering density is $\max(\min(|\mathbb{L}^{(i)}|^3, 5000), 500)$, i.e. the size of the factor's label space cubed, lower bounded at 500 and upper bounded at 5000. In the pruning step, this is reduced to $\max(\min(\frac{1}{5}|\mathbb{L}^{(i)}|^3, 1000), 100)$. In the label partitioning step, the probability threshold for computing the size of the hyper-rectangles representing the bounding regions for each track is $P_G = 0.99$. The maximum size of the label space of any single factor is set to $L_{\max} = 20$. If the constraint imposed by $L_{\max}$ cannot be satisfied with the chosen value for P_G, the partitioning routine is repeated by progressively reducing P_G by 20% until the constraint can be satisfied.

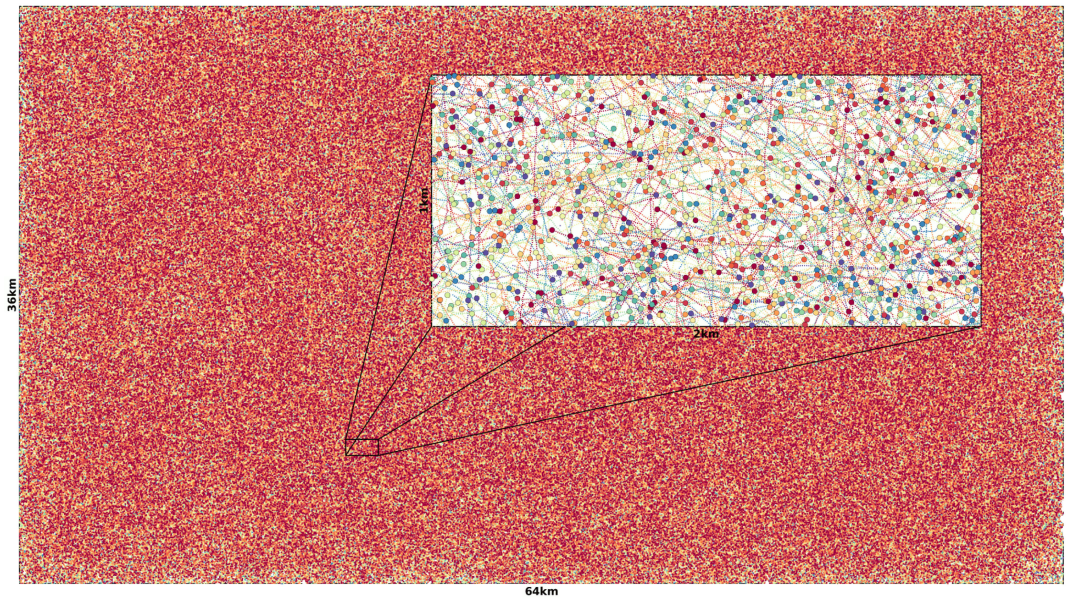


Fig. 5. Estimated tracks at time step 700 when there is a peak cardinality of 1.2 million objects. The inset is a magnified view of a 2 km by 1 km region.

The tracking algorithm was coded in C++, making use of OpenMP to implement parallel processing wherever possible. We executed the algorithm on a machine with four 16-core AMD Opteron 6376 processors (for a total of 64 physical processor cores), and 256 GB of memory. On this hardware configuration, the peak time taken to process a frame of measurements was approximately 5 minutes when the cardinality was highest, but the algorithm ran considerably faster than this at times when there were fewer objects in the scene. The peak memory usage of the algorithm was approximately 50 GB. To evaluate tracking performance, the OSPA$^{(2)}$ metric was coded in Matlab, using the "parfor" construct to parallelise some aspects of the computation. The average time taken to evaluate each point in the OSPA$^{(2)}$ curve was approximately 1 minute.

The true and estimated cardinality is shown in Fig. 3, and Fig. 4 shows OSPA$^{(2)}$ distance, as well as the traditional OSPA distance. For the OSPA calculations, the cutoff was set to $c = 50m$, the order was $p = 1$, and for the OSPA$^{(2)}$, a sliding window over the latest 50 time steps was used to evaluate each point in the curve. From the cardinality plot, it can be seen that the estimated cardinality lags behind the true cardinality during the times when new targets are being born. This is to be expected, as the measurement-driven birth model needs to consider a very large number of potential birth tracks at each scan. To avoid initiating too many false tracks, the filter delays the initiation of tracks until there is more data to confirm the presence of an object. The OSPA$^{(2)}$ plot shows increased error during the periods in which new targets are appearing, due to the delay in initiating new tracks. At other times, the error stabilises to approximately $2.5m$ per object per unit time. As we would expect, the OSPA$^{(2)}$ error is consistently higher than the OSPA error, since it penalises incorrect labelling behaviour which is not captured by the OSPA distance. Notably, the difference between the two curves is greatest during the times when the true cardinality is increasing. This is to be expected, since track initiation is arguably one of the most challenging aspects of multi-object tracking, and we can thus expect the tracker to commit more labelling errors at times when there are many new objects appearing. Fig. 5 show a snapshot of the estimated tracks produced by the proposed large-scale GLMB tracker at time 700 when the peak cardinality of approximately 1.2 million objects is reached. Two illustrative videos are also provided in supplementary materials, one showing only the measurements in time, and the other showing the estimates in time.

VII. Conclusion

We have presented an efficient and scalable implementation of the generalised labeled multi-Bernoulli filter, that is capable of estimating the trajectories of a very large number of objects simultaneously, in the order of millions per frame. The proposed method makes efficient use of the available computational resources, by decomposing large-scale tracking problems into smaller independent sub-problems. The decomposition is carried out via marginalisation of high-dimensional multi-object densities, using a technique that is shown to be optimal in the sense that it minimises the Kullback-Leibler divergence for a given partition of the object label space. This allows the algorithm to fully exploit the potential of highly parallel processing, as afforded by modern multi-core computing architectures. Due to its relatively low processing time and memory requirements, simulations show that the proposed technique is capable of tracking in excess of a million objects simultaneously, running on standard off-the-shelf computing equipment. Additionally, we have introduced a new way of using the OSPA metric to capture multi-object tracking error (rather than filtering error), and applied it to evaluate the performance of our large-scale multi-object tracker.

Appendix

A. GLMB Approximation and Factorisation

The proofs of Propositions 1 and 2, draw on the preliminary results Lemma 4, Lemma 5 and Corollary 6 below.

Lemma 4: Consider a labeled RFS density $\boldsymbol{\pi}$ on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$, and a partition $\mathfrak{L}$ of $\mathbb{L}$. The $\mathfrak{L}$-partition labeled RFS density $\boldsymbol{\pi}_{\mathfrak{L}}$ that minimises the Kullback-Leibler divergence $D_{KL}(\boldsymbol{\pi};\boldsymbol{\pi}_{\mathfrak{L}})$ is given by $\boldsymbol{\pi}_{\mathfrak{L}}=\{\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}\}_{L\in\mathfrak{L}}$, where

$$\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}(\boldsymbol{X}^{(L)})=\int\boldsymbol{\pi}\left(\mathbf{X}\right)\delta(\boldsymbol{X}-\boldsymbol{X}^{(L)}).$$

Proof: The proof follows that for the vector case [50], except that set integrals replace standard integrals. Since $\mathfrak{L}$ is a partition of $\mathbb{L}$, $\{\boldsymbol{X}^{(L)}=\boldsymbol{X}\cap(\mathbb{X}\times L)\}_{L\in\mathfrak{L}}$ is a partition of $\boldsymbol{X}$, and $\delta(\boldsymbol{X}-\boldsymbol{X}^{(L)})$ can be replaced with $\prod_{\ell\in\mathbb{L}-L}\delta\boldsymbol{X}^{(\{\ell\})}$. Expanding the Kullback-Leibler divergence between $\boldsymbol{\pi}$ and an arbitrary $\mathfrak{L}$-*partitioned* labeled RFS density $\tilde{\boldsymbol{\pi}}_{\mathfrak{L}}$ and regrouping yields $D_{KL}(\boldsymbol{\pi};\tilde{\boldsymbol{\pi}}_{\mathfrak{L}})=D_{KL}(\boldsymbol{\pi};\boldsymbol{\pi}_{\mathfrak{L}})+\sum_{L\in\mathfrak{L}}D_{KL}(\boldsymbol{\pi}_{\mathfrak{L}}^{(L)};\tilde{\boldsymbol{\pi}}_{\mathfrak{L}}^{(L)})$. The result follows by noting that the marginals $\tilde{\boldsymbol{\pi}}_{\mathfrak{L}}^{(L)}=\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}$ result in the minimum divergence.

Lemma 5: Consider a GLMB $\boldsymbol{\pi}=\{(w^{(I,c)},p^{(c)})\}_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}$ on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$, and a partition $\mathfrak{L}$ of $\mathbb{L}$. The $\mathfrak{L}$-partitioned labeled RFS density $\boldsymbol{\pi}_{\mathfrak{L}}=\{\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}\}_{L\in\mathfrak{L}}$ that minimises $D_{KL}(\boldsymbol{\pi};\boldsymbol{\pi}_{\mathfrak{L}})$, is an $\mathfrak{L}$-partitioned GLMB, with GLMB factors

$$\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}=\left\{\left(w_{\mathfrak{L},L}^{(J,c)},p_{\mathfrak{L},L}^{(c)}\right)\right\}_{(J,c)\in\mathcal{F}(L)\times\mathbb{C}}$$

$$w_{\mathfrak{L},L}^{(J,c)}=\sum_{U\in\mathcal{F}(\mathbb{L}-L)}w^{(J\cup U,c)}$$

$$p_{\mathfrak{L},L}^{(c)}(x,\ell)=1_{L}(\ell)p^{(c)}\left(x,\ell\right).$$

Proof: Using the delta form for GLMB [36],

$$\boldsymbol{\pi}\left(\boldsymbol{X}\right)=\Delta\left(\boldsymbol{X}\right)\sum_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}w^{(I,c)}\delta_{I}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\left[p^{(c)}\right]^{\boldsymbol{X}}$$

it follows from Lemma 4 that for any $\boldsymbol{X}^{(L)}\in\mathcal{F}(\mathbb{X}\times L)$,

$$\begin{aligned}\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}(\boldsymbol{X}^{(L)})&=\int\boldsymbol{\pi}\left(\boldsymbol{X}\right)\delta\left(\boldsymbol{X}-\boldsymbol{X}^{(L)}\right)\\&=\int\boldsymbol{\pi}(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})})\delta\boldsymbol{X}^{(\bar{L})}\end{aligned}$$

where $\bar{L}=\mathbb{L}-L$, and $\boldsymbol{X}^{(\bar{L})}=\boldsymbol{X}-\boldsymbol{X}^{(L)}=\boldsymbol{X}\cap(\mathbb{X}\times\bar{L})$.

Substituting for $\boldsymbol{\pi}(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})})$ gives

$$\begin{aligned}&\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}(\boldsymbol{X}^{(L)})\\&=\int\Delta\left(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})}\right)\sum_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}w^{(I,c)}\\&\quad\times\delta_{I}\left(\mathcal{L}\left(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})}\right)\right)[p^{(c)}]^{\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})}}\delta\boldsymbol{X}^{(\bar{L})}\\&=\int\Delta(\boldsymbol{X}^{(L)})\Delta\left(\boldsymbol{X}^{(\bar{L})}\right)\sum_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}w^{(I,c)}\\&\quad\times\delta_{I}\left(\mathcal{L}\left(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})}\right)\right)[p^{(c)}]^{\boldsymbol{X}^{(L)}}[p^{(c)}]^{\boldsymbol{X}^{(\bar{L})}}\delta\boldsymbol{X}^{(\bar{L})}\\&=\Delta(\boldsymbol{X}^{(L)})\sum_{(I,c)\in\mathcal{F}(\mathbb{L})\times\mathbb{C}}w^{(I,c)}[p^{(c)}]^{\boldsymbol{X}^{(L)}}\\&\quad\times\int\Delta\left(\boldsymbol{X}^{(\bar{L})}\right)\delta_{I}\left(\mathcal{L}\left(\boldsymbol{X}^{(L)}\uplus\boldsymbol{X}^{(\bar{L})}\right)\right)[p^{(c)}]^{\boldsymbol{X}^{(\bar{L})}}\delta\boldsymbol{X}^{(\bar{L})}\\&=\Delta(\boldsymbol{X}^{(L)})\sum_{J\in\mathcal{F}(L)}\sum_{U\in\mathcal{F}(\bar{L})}\sum_{c\in\mathbb{C}}w^{(J\cup U,c)}\delta_{J}(\mathcal{L}(\boldsymbol{X}^{(L)}))[p^{(c)}]^{\boldsymbol{X}^{(L)}}\\&\quad\times\int\Delta(\boldsymbol{X}^{(\bar{L})})\delta_{U}\left(\mathcal{L}\left(\boldsymbol{X}^{(\bar{L})}\right)\right)[p^{(c)}]^{\boldsymbol{X}^{(\bar{L})}}\delta\boldsymbol{X}^{(\bar{L})}\end{aligned}$$

where the last line follows from decomposing the sum over $\mathcal{F}(\mathbb{L})$, into sums over $\mathcal{F}(L)$ and its complement $\mathcal{F}(\bar{L})$.

Using Lemma 3 of [36],

$$\begin{aligned}&\int\Delta(\boldsymbol{X}^{(\bar{L})})\delta_{U}(\mathcal{L}(\boldsymbol{X}^{(\bar{L})}))[p^{(c)}]^{\boldsymbol{X}^{(\bar{L})}}\delta\boldsymbol{X}^{(\bar{L})}\\&=\sum_{H\in\mathcal{F}(\bar{L})}\delta_{U}(H)\left[\int p^{(c)}(x,\cdot)dx\right]^{H}\\&=\sum_{H\in\mathcal{F}(\bar{L})}\delta_{U}(H)=1.\end{aligned}$$

Moreover, noting that the argument of $p^{(c)}$ in $[p^{(c)}]^{\boldsymbol{X}^{(L)}}$ is restricted to $\mathbb{X}\times L$, we have $[p^{(c)}]^{\boldsymbol{X}^{(L)}}=[p_{\mathfrak{L},L}^{(c)}]^{\boldsymbol{X}^{(L)}}$, and hence the desired result.

Treating L as $\mathbb{L}$, and $\mathbb{C}(L)$ as $\mathbb{C}$ in Lemma 5 yields:

Corollary 6: Given $L\in\mathcal{F}(\mathbb{L})$, a GLMB in $\mathcal{G}_{L}$ of the form

$$\boldsymbol{\pi}^{(L)}=\left\{(w_{L}^{(I,c)},p_{L}^{(c)})\right\}_{(I,c)\in\mathcal{F}(L)\times\mathbb{C}(L)},$$

and any $S\in\mathcal{F}(\mathbb{L})$ such that $L\cap S\neq\emptyset$. The marginal

$$\boldsymbol{\pi}^{(L,S)}(\boldsymbol{X}^{(L\cap S)})=\int\boldsymbol{\pi}^{(L)}(\boldsymbol{X}^{(L)})\delta(\boldsymbol{X}^{(L)}-\boldsymbol{X}^{(S)})$$

is a GLMB in $\mathcal{G}_{L\cap S}$ given by

$$\boldsymbol{\pi}^{(L,S)}=\left\{\left(w_{L,S}^{(H,c)},p_{L,S}^{(c)}\right)\right\}_{(H,c)\in\mathcal{F}(L\cap S)\times\mathbb{C}(L)}$$

$$w_{L,S}^{(H,c)}=\sum_{W\in\mathcal{F}(L-S)}w_{L}^{(H\cup W,c)}$$

$$p_{L,S}^{(c)}(x,\ell)=1_{L\cap S}(\ell)p_{L}^{(c)}(x,\ell).$$

Proof of Proposition 1: Applying Lemma 4, the optimal approximation is given by the product of its marginals

$$\boldsymbol{\pi}_{\mathfrak{S}}\left(\boldsymbol{X}\right)=\prod_{S\in\mathfrak{S}}\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}(\boldsymbol{X}^{(S)}),$$

$$\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}(\boldsymbol{X}^{(S)})=\int\boldsymbol{\pi}_{\mathfrak{L}}\left(\boldsymbol{X}\right)\delta(\boldsymbol{X}-\boldsymbol{X}^{(S)}).$$

Since $\{\boldsymbol{X}^{(L)}\}_{L\in\mathfrak{L}}$ is a partition of $\boldsymbol{X}$, observe that $\{(\boldsymbol{X}^{(L)}-\boldsymbol{X}^{(S)})\}_{L\in\mathfrak{L}}$ is a partition of $\boldsymbol{X}-\boldsymbol{X}^{(S)}$, hence the integration can be performed over

$$\delta(\boldsymbol{X}-\boldsymbol{X}^{(S)})=\prod_{L\in\mathfrak{L}}\delta(\boldsymbol{X}^{(L)}-\boldsymbol{X}^{(S)}).$$

Substituting for $\boldsymbol{\pi}_{\mathfrak{L}}$ and regrouping yields

$$\boldsymbol{\pi}_{\mathfrak{S}}^{(S)}(\boldsymbol{X}^{(S)})=\prod_{L\in\mathfrak{L}}\boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}(\boldsymbol{X}^{(S)}),$$

$$\boldsymbol{\pi}_{\mathfrak{L},\mathfrak{S}}^{(L,S)}(\boldsymbol{X}^{(L\cap S)})=\int\boldsymbol{\pi}_{\mathfrak{L}}^{(L)}(\boldsymbol{X}^{(L)})\delta(\boldsymbol{X}^{(L)}-\boldsymbol{X}^{(S)}).$$

Applying Corollary 6 to evaluate the marginal $\boldsymbol{\pi}^{(L,S)}_{\mathfrak{L},\mathfrak{S}}$ yields the desired result.

Proof of Proposition 2: Given a partition $\mathfrak{L}_+$ of $\mathbb{L}_+$, define the corresponding partition $\mathfrak{S}$ of $\mathbb{L}$ and $\mathfrak{B}_+$ of $\mathbb{B}_+$

$$\mathfrak{S} = \{J \cap \mathbb{L} : J \in \mathfrak{L}_+\},$$
$$\mathfrak{B}_+ = \{J \cap \mathbb{B}_+ : J \in \mathfrak{L}_+\}.$$

Let $Z_+ \triangleq \biguplus_{J\in\mathfrak{L}_+} Z_+^{(J)}$, and $X_+ \triangleq \biguplus_{J\in\mathfrak{L}_+} \boldsymbol{X}_+^{(J)}$. Then the next posterior is given by

$$\boldsymbol{\pi}_{\mathfrak{L}_+,+}(\boldsymbol{X}_+|Z_+) \propto g(Z_+|\boldsymbol{X}_+)\, \boldsymbol{f}_B(\boldsymbol{X}_{B,+}) \times \int \boldsymbol{f}_S(\boldsymbol{X}_{S,+}|\boldsymbol{X}_S)\, \boldsymbol{\pi}_{\mathfrak{S}}(\boldsymbol{X}_S)\, \delta\boldsymbol{X}_S,$$
$$\boldsymbol{X}_{S,+} = \boldsymbol{X}_+ \cap (\mathbb{X}\times\mathbb{L}),$$
$$\boldsymbol{X}_{B,+} = \boldsymbol{X}_+ \cap (\mathbb{X}\times\mathbb{B}_+).$$

Noting that $\mathfrak{S}$ is a partition of $\mathbb{L}$, it follows that the transition factors into

$$\boldsymbol{f}_S(\boldsymbol{X}_{S,+}|\boldsymbol{X}_S) = \prod_{J\in\mathfrak{L}_+} \boldsymbol{f}_S(\boldsymbol{X}_{S,+}^{(J\cap\mathbb{L})}|\boldsymbol{X}_S^{(J\cap\mathbb{L})}).$$

Similarly, noting that $\mathfrak{B}_+$ is a partition of $\mathbb{B}_+$, it follows that the birth factors into

$$\boldsymbol{f}_B^{(\mathbb{B}_+)}(\boldsymbol{X}_{B,+}) = \prod_{J\in\mathfrak{L}_+} \boldsymbol{f}_B^{(J\cap\mathbb{B}_+)}(\boldsymbol{X}_{B,+}^{(J\cap\mathbb{B}_+)}).$$

Substituting

$$\boldsymbol{\pi}_{\mathfrak{S}}(\boldsymbol{X}_S) = \prod_{J\in\mathfrak{L}_+} \boldsymbol{\pi}_{\mathfrak{S}}^{(J\cap\mathbb{L})}(\boldsymbol{X}_S^{(J\cap\mathbb{L})}),$$
$$g(Z_+|\boldsymbol{X}_+) = \prod_{J\in\mathfrak{L}_+} g(Z_+^{(J)}|\boldsymbol{X}_+^{(J)}),$$

yields

$$\boldsymbol{\pi}_{\mathfrak{L}_+,+}(\boldsymbol{X}_+|Z_+) = \prod_{J\in\mathfrak{L}_+} \pi^{(J)}_{\mathfrak{L}_+,+}(Z_+^{(J)}|\boldsymbol{X}_+^{(J)}),$$
$$\boldsymbol{\pi}^{(J)}_{\mathfrak{L}_+,+}(Z_+^{(J)}|\boldsymbol{X}_+^{(J)}) = g(Z_+^{(J)}|\boldsymbol{X}_+^{(J)})\boldsymbol{f}_B^{(J\cap\mathbb{B}_+)}(\boldsymbol{X}_{B,+}^{(J\cap\mathbb{B}_+)}) \times \int \boldsymbol{f}_S(\boldsymbol{X}_{S,+}^{(J\cap\mathbb{L})}|\boldsymbol{X}_S^{(J\cap\mathbb{L})})\boldsymbol{\pi}_{\mathfrak{S}}(\boldsymbol{X}_S^{(J\cap\mathbb{L})})\delta\boldsymbol{X}_S^{(J\cap\mathbb{L})}.$$

Applying Proposition 1 in [38], for each $J \in \mathfrak{L}_+$ gives the final result.

B. Proof of Proposition 3 (Metric for Tracks)

Firstly, since $d^{(c)}(\cdot,\cdot) \geq 0$ and $|\mathcal{D}_x \cup \mathcal{D}_y| \geq 0$, all terms in the summation over $t \in \mathcal{D}_x \cup \mathcal{D}_y$ are clearly non-negative. Hence $\tilde{d}^{(c)}(\cdot,\cdot)$ satisfies metric property P1.

Second, $\tilde{d}^{(c)}(x,y) = 0$ if and only if $x = y = \emptyset$, or $d^{(c)}(\{x(t)\},\{y(t)\}) = 0$ for all $t \in \mathcal{D}_x \cup \mathcal{D}_y$. Since $d^{(c)}(\cdot,\cdot)$ is a metric, $d^{(c)}(\{x(t)\},\{y(t)\}) = 0 \Leftrightarrow \{x(t)\} = \{y(t)\}$. Hence $\tilde{d}^{(c)}(x,y) = 0 \Leftrightarrow x = y$, satisfying metric property P2.

Third, since $d^{(c)}(\cdot,\cdot)$ is a metric, and $\mathcal{D}_x \cup \mathcal{D}_y = \mathcal{D}_y \cup \mathcal{D}_x$, all terms in (35) are symmetric in their arguments. Hence $\tilde{d}^{(c)}(\cdot,\cdot)$ satisfies metric property P3.

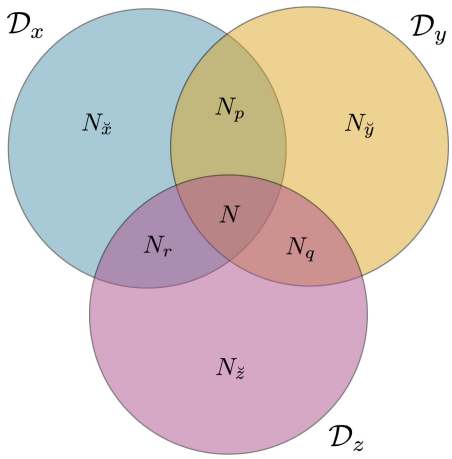
It remains to verify metric property P4 (the triangle inequality), which is accomplished via induction. Since $d^{(c)}(\cdot,\cdot)$ is a metric, it is clear that the distance between the tracks at a single time index 1 (representing time t_1), satisfies the triangle inequality. Let us assume that the distance between the tracks over time indices $\{1,2,\ldots,k\}$ (representing $\{t_1,t_2,\ldots,t_k\}$), satisfies the triangle inequality. We now proceed to show that the distance between the tracks over time indices $\{1,2,\ldots,k,k+1\}$ also satisfies the triangle inequality.

When at least one of the sets $\mathcal{D}_x \cup \mathcal{D}_y$, $\mathcal{D}_y \cup \mathcal{D}_z$, $\mathcal{D}_z \cup \mathcal{D}_x$ is empty, the triangle inequality for tracks over time indices $1,2,\ldots,k,k+1$ can be easily verified, since $d^{(c)}(\cdot,\cdot)$ is a metric. Hence, we consider the case where $\mathcal{D}_x \cup \mathcal{D}_y$, $\mathcal{D}_y \cup \mathcal{D}_z$, $\mathcal{D}_z \cup \mathcal{D}_x$ are all non-empty.

Some notations are needed for compactness. Let us denote the cardinalities of the basic sets in $\mathcal{D}_x \cup \mathcal{D}_y \cup \mathcal{D}_z$ by

$$N \triangleq |\mathcal{D}_x \cap \mathcal{D}_y \cap \mathcal{D}_z|, \tag{38}$$
$$N_{\check{x}} \triangleq |\mathcal{D}_x - \mathcal{D}_y - \mathcal{D}_z|, \quad N_p \triangleq |\mathcal{D}_x \cap \mathcal{D}_y - \mathcal{D}_z|, \tag{39}$$
$$N_{\check{y}} \triangleq |\mathcal{D}_y - \mathcal{D}_z - \mathcal{D}_x|, \quad N_q \triangleq |\mathcal{D}_y \cap \mathcal{D}_z - \mathcal{D}_x|, \tag{40}$$
$$N_{\check{z}} \triangleq |\mathcal{D}_z - \mathcal{D}_x - \mathcal{D}_y|, \quad N_r \triangleq |\mathcal{D}_z \cap \mathcal{D}_x - \mathcal{D}_y|, \tag{41}$$

as illustrated in the diagram below.



Furthermore, we adopt the following abbreviations:

$$S \triangleq N + N_p + N_q + N_r, \tag{42}$$
$$T \triangleq S + |\mathcal{D}_x \cup \mathcal{D}_y \cup \mathcal{D}_z| = 2S + N_{\check{x}} + N_{\check{y}} + N_{\check{z}}, \tag{43}$$
$$P \triangleq |\mathcal{D}_x \cup \mathcal{D}_y| = S + N_{\check{x}} + N_{\check{y}}, \tag{44}$$
$$Q \triangleq |\mathcal{D}_y \cup \mathcal{D}_z| = S + N_{\check{y}} + N_{\check{z}}, \tag{45}$$
$$R \triangleq |\mathcal{D}_z \cup \mathcal{D}_x| = S + N_{\check{x}} + N_{\check{z}}, \tag{46}$$
$$p \triangleq \sum_{t\in\mathcal{D}_x\cup\mathcal{D}_y} d^{(c)}(\{x(t)\},\{y(t)\}), \tag{47}$$
$$p' \triangleq d^{(c)}(\{x(k+1)\},\{y(k+1)\}), \tag{48}$$
$$q \triangleq \sum_{t\in\mathcal{D}_y\cup\mathcal{D}_z} d^{(c)}(\{y(t)\},\{z(t)\}), \tag{49}$$
$$q' \triangleq d^{(c)}(\{y(k+1)\},\{z(k+1)\}) \tag{50}$$

$$r \triangleq \sum_{t\in\mathcal{D}_z\cup\mathcal{D}_x} d^{(c)}\left(\{z(t)\},\{x(t)\}\right), \tag{51}$$

$$r' \triangleq d^{(c)}\left(\{z\left(k+1\right)\},\{x\left(k+1\right)\}\right). \tag{52}$$

The sum p over (non-empty) $\mathcal{D}_x \cup \mathcal{D}_y$ is decomposed into:

- $\hat{p}_N$ consisting of N terms on $(\mathcal{D}_x \cap \mathcal{D}_y \cap \mathcal{D}_z)$;
- $\hat{p}_{N_p}$ consisting of N_p terms on $(\mathcal{D}_x \cap \mathcal{D}_y - \mathcal{D}_z)$; and
- $(N_q + N_r + N_{\check{x}} + N_{\check{y}})c$, since $d(\{x(k)\},\{y(k)\}) = c$ on $(\mathcal{D}_x \cup \mathcal{D}_y) - (\mathcal{D}_x \cap \mathcal{D}_y)$.

Similar decompositions also apply to q, and r. Hence,

$$p \triangleq \hat{p}_N + \hat{p}_{N_p} + \left(N_q + N_r + N_{\check{x}} + N_{\check{y}}\right)c,$$

$$q \triangleq \hat{q}_N + \hat{q}_{N_q} + \left(N_r + N_p + N_{\check{y}} + N_{\check{z}}\right)c,$$

$$r \triangleq \hat{r}_N + \hat{r}_{N_r} + \left(N_p + N_q + N_{\check{z}} + N_{\check{x}}\right)c.$$

The following bounds are required for the proof

$$\hat{p}_N \le \hat{q}_N + \hat{r}_N, \tag{53}$$

$$p \le \hat{p}_N + (P-N)\,c \le Pc, \tag{54}$$

$$q \ge \hat{q}_N + \left(N_r + N_p + N_{\check{y}} + N_{\check{z}}\right)c \triangleq \hat{q}_N + Q^{\circledcirc}c, \tag{55}$$

$$r \ge \hat{r}_N + \left(N_p + N_q + N_{\check{z}} + N_{\check{x}}\right)c \triangleq \hat{r}_N + R^{\circledcirc}c, \tag{56}$$

Note that: the triangle inequality (53) holds because $\hat{p}_N$, $\hat{q}_N$, $\hat{r}_N$ are, respectively, sums of distances between x and y, y and z, z and x, over all time indices in $\mathcal{D}_x \cap \mathcal{D}_y \cap \mathcal{D}_z$; (54) holds because $\hat{p}_{N_p} \le N_p c$, $N_p + N_q + N_r + N_{\check{x}} + N_{\check{y}} = P - N$, and $\hat{p}_N \le Nc$; and (55), (56) hold because $\hat{q}_{N_q}, \hat{r}_{N_r} \ge 0$.

The following identities (follows directly from the definitions)

$$P + Q = T + N_{\check{y}}, \tag{57}$$

$$= R + S + 2N_{\check{y}}, \tag{58}$$

$$Q + R = T + N_{\check{z}}, \tag{59}$$

$$R + P = T + N_{\check{x}}, \tag{60}$$

$$N + Q^{\circledcirc} + R^{\circledcirc} - Q = N_p + N_{\check{x}} + N_{\check{z}}, \tag{61}$$

$$N + Q^{\circledcirc} + R^{\circledcirc} - P = N_p + 2N_{\check{z}}, \tag{62}$$

are also required for the proof. Note that so far, all of the variables we have defined are non-negative.

Adopting the above notation, the properties of $d^{(c)}(\cdot,\cdot)$ and the triangle inequality for $\tilde{d}(\cdot,\cdot)$ can be expressed as

$$c \ge p', q', r', \tag{63}$$

$$r' + q' - p' \ge 0, \tag{64}$$

$$\frac{r}{R} + \frac{q}{Q} \ge \frac{p}{P} \text{ or equivalently } PQr + RPq - QRp \ge 0. \tag{65}$$

We need to prove that the triangle inequality holds for the following three cases (note that the result holds trivially when $\{x(k+1)\} = \{y(k+1)\} = \{z(k+1)\} = \emptyset$):

(i) $\{x(k+1)\} = \{y(k+1)\} = \emptyset$, and $\{z(k+1)\} \ne \emptyset$, i.e.

$$\frac{r+c}{R+1} + \frac{q+c}{Q+1} \ge \frac{p}{P}.$$

(ii) $\{z(k+1)\} = \emptyset$, and $\{x(k+1)\} \ne \emptyset$ or $\{y(k+1)\} \ne \emptyset$, i.e.

$$\frac{r}{R} + \frac{q+c}{Q+1} \ge \frac{p+c}{P+1} \text{ or } \frac{r+c}{R+1} + \frac{q}{Q} \ge \frac{p+c}{P+1}.$$

(iii) At least two of $\{x(k+1)\}$, $\{y(k+1)\}$ and $\{z(k+1)\}$ are non-empty, i.e.

$$\frac{r+r'}{R+1} + \frac{q+q'}{Q+1} \ge \frac{p+p'}{P+1}.$$

For case (i)

$$\begin{aligned}
&\frac{r+c}{R+1} + \frac{q+c}{Q+1} - \frac{p}{P} \\
&= \frac{P\left(Q+1\right)r + P\left(Q+1\right)c}{P\left(Q+1\right)\left(R+1\right)} + \frac{\left(R+1\right)Pq + \left(R+1\right)Pc}{P\left(Q+1\right)\left(R+1\right)} \\
&\quad - \frac{\left(Q+1\right)\left(R+1\right)p}{P\left(Q+1\right)\left(R+1\right)} \\
&= \frac{P\left(Q+1\right)r + \left(R+1\right)Pc}{P\left(Q+1\right)\left(R+1\right)} + \frac{Pc + \left(Q+R+1\right)Pc}{P\left(Q+1\right)\left(R+1\right)} \\
&\quad - \frac{\left(QR+Q+R+1\right)p}{P\left(Q+1\right)\left(R+1\right)} \\
&= \frac{PQr + RPq - QRp + Pr + Pq + Pc}{P\left(Q+1\right)\left(R+1\right)} \\
&\quad + \frac{\left(Q+R+1\right)Pc - \left(Q+R+1\right)p}{P\left(Q+1\right)\left(R+1\right)} \\
&\ge \frac{P\left(r+q+c\right) + \left(Q+R+1\right)\left(Pc-p\right)}{P\left(Q+1\right)\left(R+1\right)} \ge 0,
\end{aligned}$$

where we used the triangle inequality (65) and the bound $Pc \ge p$ from (54).

For case (ii)

$$\begin{aligned}
&\frac{r}{R} + \frac{q+c}{Q+1} - \frac{p+c}{P+1} \\
&= \frac{\left(P+1\right)\left(Q+1\right)r}{\left(P+1\right)\left(Q+1\right)R} \\
&\quad + \frac{R\left(P+1\right)q + R\left(P+1\right)c - \left(Q+1\right)Rp - \left(Q+1\right)Rc}{\left(P+1\right)\left(Q+1\right)R} \\
&= \frac{\left(PQ+P+Q+1\right)r}{\left(P+1\right)\left(Q+1\right)R} \\
&\quad + \frac{R\left(P+1\right)q - \left(Q+1\right)Rp + R\left(P+1\right)c - \left(Q+1\right)Rc}{\left(P+1\right)\left(Q+1\right)R} \\
&= \frac{\left(PQ+P+Q+1\right)r + RPq + Rq - QRpc}{\left(P+1\right)\left(Q+1\right)R} \\
&\quad + \frac{\left(P-Q\right)Rc - Rp}{\left(P+1\right)\left(Q+1\right)R} \\
&= \frac{PQr + RPq - QRp + \left(P+Q+1\right)r + Rq}{\left(P+1\right)\left(Q+1\right)R} \\
&\quad + \frac{\left(P-Q\right)Rc - Rp}{\left(P+1\right)\left(Q+1\right)R} \\
&\ge \frac{\left(P+Q+1\right)r + Rq - Rp + \left(P-Q\right)Rc}{\left(P+1\right)\left(Q+1\right)R}
\end{aligned}$$

where the last line follows from the triangle inequality (65). Using the bounds (54), (55), (56) for p, q, and r, and the identity $P+Q=R+S+2N_{\check{y}}$ from (58), we have

$$\begin{aligned}
&(P+Q+1)\,r+Rq-Rp+(P-Q)\,Rc\\
&\geq (R+S+2N_{\check{y}}+1)\left[\hat{r}_N+R^{\circledcirc}c\right]+R\left[\hat{q}_N+Q^{\circledcirc}c\right]\\
&-R\left[\hat{p}_N+(P-N)\,c\right]+(P-Q)\,Rc\\
&=\underbrace{\begin{array}{l}+R\hat{r}_N\\+R\hat{q}_N\\-R\hat{p}_N\end{array}}_{0}\quad\underbrace{\begin{array}{l}+(S+2N_{\check{y}}+1)\,\hat{r}_N\\\\\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+R^{\circledcirc}(R+S+2N_{\check{y}}+1)\,c\\+Q^{\circledcirc}Rc\\-(P-N)\,Rc+(P-Q)\,Rc\end{array}}_{+R^{\circledcirc}Rc+Q^{\circledcirc}Rc+(N-Q)\,Rc}\\
&\geq \left[R^{\circledcirc}+Q^{\circledcirc}+N-Q\right]Rc\\
&=\left[N_p+N_{\check{x}}+N_{\check{z}}\right]Rc\geq 0,
\end{aligned}$$

where we used $\hat{r}_N+\hat{q}_N-\hat{p}_N\geq 0$ from (53) and $R^{\circledcirc}+Q^{\circledcirc}+N-Q=N_p+N_{\check{x}}+N_{\check{z}}$ from (61).

For case (iii)

$$\begin{aligned}
&\frac{r+r'}{R+1}+\frac{q+q'}{Q+1}-\frac{p+p'}{P+1}\\
&=\frac{(P+1)\,(Q+1)\,(r+r')}{(P+1)\,(Q+1)\,(R+1)}+\frac{(R+1)\,(P+1)\,(q+q')}{(P+1)\,(Q+1)\,(R+1)}\\
&-\frac{(Q+1)\,(R+1)\,(p+p')}{(P+1)\,(Q+1)\,(R+1)}\\
&=\frac{(PQ+P+Q+1)\,r}{(P+1)\,(Q+1)\,(R+1)}+\frac{(RP+R+P+1)\,q}{(P+1)\,(Q+1)\,(R+1)}\\
&-\frac{(QR+Q+R+1)\,p}{(P+1)\,(Q+1)\,(R+1)}+\frac{(PQ+P+Q+1)\,r'}{(P+1)\,(Q+1)\,(R+1)}\\
&+\frac{(RP+R+P+1)\,q'}{(P+1)\,(Q+1)\,(R+1)}-\frac{(QR+Q+R+1)\,p'}{(P+1)\,(Q+1)\,(R+1)}\\
&=\frac{(PQr+RPq-QRp)+(r'+q'-p')}{(P+1)\,(Q+1)\,(R+1)} \qquad (66)\\
&+\frac{(P+Q+1)\,r+(R+P+1)\,q-(Q+R+1)\,p}{(P+1)\,(Q+1)\,(R+1)} \qquad (67)\\
&+\frac{(PQ+P+Q)\,r'+(RP+R+P)\,q'-(QR+Q+R)\,p'}{(P+1)\,(Q+1)(R+1)}. \qquad (68)
\end{aligned}$$

Note that (66) ≥ 0 from the triangle inequalities (65) and (64). It remains to be shown that (67) + (68) ≥ 0.

Into (67), we substitute the three identities $P+Q=T+N_{\check{y}}$, $Q+R=T+N_{\check{z}}$ and $R+P=T+N_{\check{x}}$, from (57), (59) and (60) respectively. We also use the upper/lower bounds on p, q, and r, from (54), (55) and (56) respectively. This yields the following expression

$$\begin{aligned}
&(P+Q+1)\,r+(R+P+1)\,q-(Q+R+1)\,p\\
&\geq (T+N_{\check{y}}+1)\left[\hat{r}_N+R^{\circledcirc}c\right]+(T+N_{\check{x}}+1)\left[\hat{q}_N+Q^{\circledcirc}c\right]\\
&-(T+N_{\check{z}}+1)\left[\hat{p}_N+(P-N)\,c\right]\\
&=\underbrace{\begin{array}{l}+(T+1)\,\hat{r}_N\\+(T+1)\,\hat{q}_N\\-(T+1)\,\hat{p}_N\end{array}}_{0}\quad\underbrace{\begin{array}{l}+N_{\check{y}}\hat{r}_N\\+N_{\check{x}}\hat{q}_N\\-N_{\check{z}}\hat{p}_N\end{array}}_{-N_{\check{z}}Nc}\quad\underbrace{\begin{array}{l}+(T+1)\,R^{\circledcirc}c\\+(T+1)\,Q^{\circledcirc}c\\-(T+1)\,(P-N)\,c\end{array}}_{+(T+1)(N_p+2N_{\check{z}})\,c}\quad\underbrace{\begin{array}{l}+N_{\check{y}}R^{\circledcirc}c\\+N_{\check{x}}Q^{\circledcirc}c\\-N_{\check{z}}\,(P-N)\,c\end{array}}_{-N_{\check{z}}Pc+N_{\check{z}}Nc}\\
&\geq \left[(T+1)\,(N_p+2N_{\check{z}})-N_{\check{z}}P\right]c\\
&\geq \left[(T+1)\,N_{\check{z}}-N_{\check{z}}P\right]c\\
&=(S+N_{\check{z}}+1)\,N_{\check{z}}c,
\end{aligned}$$

where we used $\hat{r}_N+\hat{q}_N-\hat{p}_N\geq 0$ from (53), $N_{\check{y}}\hat{r}_N\geq 0$, $N_{\check{x}}\hat{q}_N\geq 0$, $-\hat{p}_N\geq -Nc$ from (54), $Q^{\circledcirc}+R^{\circledcirc}-P+N=N_p+2N_{\check{z}}$ from (62), $N_{\check{y}}R^{\circledcirc}c\geq 0$, $N_{\check{x}}Q^{\circledcirc}c\geq 0$, $N_p+2N_{\check{z}}\geq N_{\check{z}}$, and $T-P=S+N_{\check{z}}$ from (43) and (44).

For (68), note from (57), (59) and (60), that $P+Q=T+N_{\check{y}}$, $Q+R=T+N_{\check{z}}$ and $R+P=T+N_{\check{x}}$. Additionally,

$$\begin{aligned}
PQ&=(S+N_{\check{x}}+N_{\check{y}})\,(S+N_{\check{y}}+N_{\check{z}})\\
&=SS+SN_{\check{x}}+2SN_{\check{y}}+SN_{\check{z}}\\
&\quad+N_{\check{x}}N_{\check{y}}+N_{\check{y}}N_{\check{z}}+N_{\check{z}}N_{\check{x}}+N_{\check{y}}N_{\check{y}},\\
QR&=(S+N_{\check{y}}+N_{\check{z}})\,(S+N_{\check{x}}+N_{\check{z}})\\
&=SS+SN_{\check{x}}+SN_{\check{y}}+2SN_{\check{z}}\\
&\quad+N_{\check{x}}N_{\check{y}}+N_{\check{y}}N_{\check{z}}+N_{\check{z}}N_{\check{x}}+N_{\check{z}}N_{\check{z}},\\
RP&=(S+N_{\check{x}}+N_{\check{z}})\,(S+N_{\check{x}}+N_{\check{y}})\\
&=SS+2SN_{\check{x}}+SN_{\check{y}}+SN_{\check{z}}\\
&\quad+N_{\check{x}}N_{\check{y}}+N_{\check{y}}N_{\check{z}}+N_{\check{z}}N_{\check{x}}+N_{\check{x}}N_{\check{x}}.
\end{aligned}$$

Hence, expanding (68) results in (69) shown at the bottom of this page, where we used the triangle inequality (64) for p', q', r'.

Finally, since $c\geq p'$, it follows that (67) + (68) ≥ 0.

$$\begin{aligned}
&(PQ+P+Q)\,r'+(RP+R+P)\,q'-(QR+Q+R)\,p'\\
&=\underbrace{\begin{array}{l}+SSr'\\+SSq'\\-SSp'\end{array}}_{0}\quad\underbrace{\begin{array}{l}+SN_{\check{x}}r'\\+2SN_{\check{x}}q'\\-SN_{\check{x}}p'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+2SN_{\check{y}}r'\\+SN_{\check{y}}q'\\-SN_{\check{y}}p'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+SN_{\check{z}}r'\\+SN_{\check{z}}q'\\-2SN_{\check{z}}p'\end{array}}_{-SN_{\check{z}}p'}\quad\underbrace{\begin{array}{l}+N_{\check{x}}N_{\check{y}}r'\\+N_{\check{x}}N_{\check{y}}q'\\-N_{\check{x}}N_{\check{y}}p'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+N_{\check{y}}N_{\check{z}}r'\\+N_{\check{y}}N_{\check{z}}q'\\-N_{\check{y}}N_{\check{z}}p'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+N_{\check{z}}N_{\check{x}}r'\\+N_{\check{z}}N_{\check{x}}q'\\-N_{\check{z}}N_{\check{x}}p'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+N_{\check{y}}N_{\check{y}}r'\\+N_{\check{x}}N_{\check{x}}q'\\-N_{\check{z}}N_{\check{z}}p'\end{array}}_{-N_{\check{z}}N_{\check{z}}p'}\quad\underbrace{\begin{array}{l}+Tr'\\+Tq'\\-Tp'\end{array}}_{+0}\quad\underbrace{\begin{array}{l}+N_{\check{y}}r'\\+N_{\check{x}}q'\\-N_{\check{z}}p'\end{array}}_{-N_{\check{z}}p'}\\
&=-(S+N_{\check{z}}+1)\,N_{\check{z}}p', \qquad (69)
\end{aligned}$$

It is possible to extend the base-distance (33) to the case where the terms of the sum in (33) are raised to the power of q, and take the qth root of the resulting sum.

REFERENCES

[1] S. Blackman, R. Popoli, *Design and Analysis of Modern Tracking Systems*, Norwood, MA, USA: Artech House, 1999.

[2] Y. B.-Shalom, X. Rong Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Bradford, U.K.: YBS Publishing, 1995.

[3] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Norwood, MA, USA: Artech House, 2007.

[4] B. A. Jones, D. S. Bryant, B.-T. Vo, and B.-N. Vo, "Challenges of multi-target tracking for space situational awareness," in *Proc. 18th Int. Conf. Inf. Fusion*, Washington D.C., Jul. 2015, pp. 1278–1285.

[5] H. Klinkrad, "Space debris: Models and risk analysis," Berlin, Germany: Springer, 2006.

[6] Committee for the Assessment of the U.S. Air Force' s Astrodynamic Standards; Aeronautics and Space Engineering Board; Division on Engineering and Physical Sciences; National Research Council, *Continuing Kepler' s Quest: Assessing Air Force Space Command' s Astrodynamics Standards*, Washington, DC, USA: The National Academies Press, 2012.

[7] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *Proc. Eur. Conf. Computer Vision*, Springer, Berlin, Heidelberg, Sep. 2010, pp. 186–199.

[8] T. Pollard and M. Antone, "Detecting and tracking all moving objects in wide-area aerial video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. Workshops*, Jun. 2012, pp. 15–22.

[9] J. Prokaj, X. Zhao, and G. Medioni, "Tracking many vehicles in wide area aerial surveillance," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn. Workshops*, Jun. 2012, pp. 37–43.

[10] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, "Multiple hypothesis tracking for cluttered biological image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2736–3750, Nov. 2013.

[11] E. Meijering, O. Dzyubachyk and I. Smal, "Methods for cell and particle tracking," in *Methods in Enzymology*, vol. 504, New York, NY, USA: Academic, 2012, pp. 183–200.

[12] J. K. Uhlmann, "Algorithms for multiple-target tracking," *Amer. Scientist*, vol. 80, no. 2, pp. 128–141, 1992.

[13] J. B. Collins and J. K. Uhlmann, "Efficient gating in data association with multivariate Gaussian distributed states," *IEEE Trans. Aerosp. Electron. Syst*, vol. 28, no. 3, pp. 909–916, Jul. 1992.

[14] D. Musicki, B. F. La Scala, and R. J. Evans, "Multi-target tracking in clutter without measurement assignment." in *Proc. 43 rd IEEE Conf. Decis. Control*, 2004, pp. 716–721.

[15] H. Wang, T. Kirubarajan, and Y. Bar-Shalom, "Precision large scale air traffic surveillance using IMM/assignment estimators," *IEEE Trans. Aerosp. Electron. Syst,*, vol. 35, no. 1, pp. 255–266, Jan. 1999.

[16] M. Betke, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and T. H. Kunz, "Tracking large variable numbers of objects in clutter," in *Proc. IEEE Conf. Computer Vision Pattern Recogn.*, 2007.

[17] B.-N. Vo, B.-T. Vo, S. Reuter, Q. Lam, and K Dietmayer, "Towards large scale multi-target tracking," in *Proc. Int. Soc. Opt. Photon.: Sensors Syst. Space Appl. VII*, 2014, vol. 9085, Art. no. 90850W.

[18] J. Olofsson, C. Veibäck, and G. Hendeby, "Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter," in *Proc. 20th Int. Conf. Inf. Fusion*, Xi'an, China, Jul. 2017.

[19] D. E. Clark, S. J. Julier, R. Mahler, and B. Ristic, "Robust multi-object sensor fusion with unknown correlations," in *Proc. Sens. Signal Process. Def.*, Sep. 2010, pp. 1–5.

[20] M. Uney, D. E. Clark, and S. J. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 521–531, Jun. 2013.

[21] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 508–520, Jun. 2013.

[22] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and R. Mahler, "Distributed fusion of multitarget densities and consensus PHD/CPHD filters," in *Proc. SPIE Def., Security Sens.*, Baltimore, MD, USA, vol. 9474, 2015.

[23] M. B. Guldogan, "Consensus Bernoulli filter for distributed detection and tracking using multi-static doppler shifts," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 672–676, Jun. 2014.

[24] W. Yi, M. Jiang, R. Hoseinnezhad, and B. Wang, "Distributed multisensor fusion using generalised multi-Bernoulli densities," *IET Radar, Sonar Navig.*, vol. 11, no. 3, pp. 434–443, Mar. 2016.

[25] B. L. Wang, W. Yi, R. Hoseinnezhad, S. Q. Li, L. J. Kong, and X. B. Yang, "Distributed fusion with multi-Bernoulli filter based on generalized covariance intersection," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 242–255, Jan. 2017.

[26] T. C. Li, J. M. Corchado, and S. D. Sun, "On generalized covariance intersection for distributed PHD filtering and a simple but better alternative," in *Proc. IEEE Int. Fusion Conf.*, 2017, pp. 1–8.

[27] F. Meyer *et al.*, "Message passing algorithms for scalable multitarget tracking," in *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[28] C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci, "Robust fusion for multisensor multiobject tracking," *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 640–644, 2018.

[29] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[30] C. Fantacci, and F. Papi, "Scalable multisensor multitarget tracking using the marginalized-GLMB density," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 863–867, Jun. 2016.

[31] X. Wang, A. K. Gostar, T. Rathnayake, B. Xu, and A. B.-Hadiashar, "Centralized multiple-view sensor fusion using labeled multi-Bernoulli filters," *Signal Process.*, vol. 150, pp. 75–84, 2018.

[32] S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, "Robust distributed fusion with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 278–293, Jan. 2018.

[33] M. Üney, J. Houssineau, E. Delande, S. J. Julier, and D. E. Clark, "Fusion of finite-set distributions: Pointwise consistency and global cardinality," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2759–2773, Dec. 2019.

[34] S. Li, G. Battistelli, L. Chisci, W. Yi, B. Wang, and L. Kong, "Computationally efficient multi-agent multi-object tracking with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 260–275, Jan. 2019.

[35] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224-1245, Oct. 2005.

[36] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460-3475, Jul. 2013.

[37] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6554–6567, Dec. 2014.

[38] B.-N. Vo, B.-T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.

[39] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3452–3457, Jul. 2011.

[40] T. Vu and R. Evans, "A new performance metric for multiple target tracking based on optimal subpattern assignment," in *Proc. 17th Int. Conf. Inf. Fusion*, Salamanca, Spain, Jul. 2014.

[41] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[42] M. Beard, B.-T. Vo, and B.-N. Vo, "OSPA$^{(2)}$: Using the OSPA metric to evaluate multi-target tracking performance," in *Proc. Int. Conf. Control, Automat. Inf. Sci.*, Chang Mai, Thailand, Oct. 2017.

[43] M. Beard, B.-T. Vo, and B.-N. Vo, "Performance evaluation for large-scale multi-target tracking algorithms," in *Proc. 21st Int. Conf. Inf. Fusion*, Cambridge, UK, Jul. 2018.

[44] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 1984, vol. 14, pp. 47–57.

[45] Y. Manolopoulos, A. Nanopoulos, and Y. Theodoridis, *R-Trees: Theory and Applications*, Berlin, Germany: Springer, 2010.

[46] A. Zomorodian and H. Edelsbrunner, "Fast software for box intersections," in *Proc. 16th Annual Symp. Comput. Geometry*, ACM, 2000, pp. 129–138.

[47] J. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[48] J. Bento, "A metric for sets of trajectories that is practical and mathematically consistent," 2016, *arXiv:1601.03094*.

[49] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms," 2016, *arXiv:1605.01177*.

[50] J. Cardoso, "Dependence, correlation and Gaussianity in independent component analysis," *J. Mach. Learn. Res.*, vol. 4, pp. 1177–1203, 2003.