

Distributed Fusion of PHD Filters via Exponential Mixture Densities

Murat Üney, *Member, IEEE*, Daniel E. Clark, *Member, IEEE*, Simon J. Julier, *Member, IEEE*,

Abstract—In this paper, we consider the problem of Distributed Multi-sensor Multi-target Tracking (DMMT) for networked fusion systems. Many existing approaches for DMMT use multiple hypothesis tracking and track-to-track fusion. However, there are two difficulties with these approaches. First, the computational costs of these algorithms can scale factorially with the number of hypotheses. Second, consistent optimal fusion, which does not double count information, can only be guaranteed for highly constrained network architectures which largely undermine the benefits of distributed fusion.

In this paper, we develop a consistent approach for DMMT by combining a generalised version of Covariance Intersection, based on Exponential Mixture Densities (EMDs), with Random Finite Sets (RFS). We first derive explicit formulae for the use of EMDs with RFSs. From this, we develop expressions for the probability hypothesis density filters. This approach supports DMMT in arbitrary network topologies through local communications and computations. We implement this approach using Sequential Monte Carlo techniques and demonstrate its performance in simulations.

Index Terms—Multi-object filtering, PHD, CPHD, multi-sensor fusion, distributed fusion, exponential mixture density, covariance intersection, multi-sensor multi-target tracking, wireless sensor networks.

I. INTRODUCTION

BECAUSE of its practical importance, Distributed Multi-sensor Multi-target Tracking (DMMT) has become increasingly important. In applications that range from traffic monitoring to battlefield surveillance, networks of multiple sensing systems are used to track the trajectories of multiple targets over time. Scalability, flexibility, robustness and fault-tolerance are often demanded. Therefore, rather than have centralised fusion in which there is a single point of failure, there is a great deal of interest in algorithms and techniques that can fuse information throughout the entire network, leading to DMMT. However, there two difficult challenges that must be overcome: multi-target tracking and distributed information fusion.

The first challenge, multi-target tracking when the number of targets is unknown, is widely recognised to be extremely

difficult [1]. Perhaps the first work in the area was Reid's pioneering work in Multiple Hypothesis Tracking (MHT) [2]. The idea underlying MHT is to enumerate all the hypotheses that describe all potential evolutions of targets and all possible associations of measurements with targets. However, the number of hypotheses grows factorially over time. To overcome these difficulties, numerous algorithms have been proposed [3]. Many approaches use a variety of track pruning, in which only a fixed number of most likely hypotheses are maintained. Recently, there has been a growing interest in reversible data association methods, in which association decisions can be revised as more information becomes available. However, because of the computational complexity typically only a single estimate over the association decisions — typically the maximum a posteriori one — is maintained [4].

The second challenge is to fuse the information from different fusion systems together in a consistent manner. In principle, this can be achieved by maintaining the marginal and joint probabilities of the distributions in the different fusion systems [5]. However, unless the network is synchronous and tree-connected, these quantities can only be computed if an oracle continuously monitors the entire state of the network [6]. This requirement undermines the potential advantages of flexibility, scalability and robustness of distributed systems.

Existing solutions for DMMT combine both techniques together in an unmodified fashion [5]. Multi-target tracking algorithms are run on each fusion system separately and yield a set of tracks. Track-to-track fusion algorithms are used to construct associations between the different tracks in the different sensing systems. Once tracks have been associated, the state from one track is fused with that of another using a distributed fusion scheme. However, these methods suffer from the scalability and fragility of MHT, and the limitations imposed by optimal distributed fusion architectures.

In this paper, we propose an approach to DMMT which addresses both of the aforementioned challenges. Our approach is to generalise the formulation of a suboptimal distributed fusion algorithm known as Covariance Intersection (CI) developed by Uhlmann [7] within the multi-object probabilistic framework developed by Mahler [8]. Mahler proposed a generalisation of CI based on Exponential Mixture Densities (EMDs) of random finite set (RFS) distributions [9]. In this work, we derive the forms of EMDs and develop algorithms for distributed multi-object filtering. Preliminary mathematical results on RFS EMDs can be found in [10] and an implementation strategy is presented in [11]. Selection of the weight parameter is considered in [12]. In this paper, we provide a full account of

Manuscript received 01 September, 2012. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) (Grant numbers EP/H011990/1 and EP/J015180/1) and the MOD University Defence Research Centre on Signal Processing (UDRC). Daniel E. Clark is a RA Eng./EPSRC Research Fellow.

Murat Üney and Daniel E. Clark are with Heriot-Watt University (HWU), Edinburgh, in Joint Research Initiative on Signal and Image Processing within the Edinburgh Research Partnership. Murat Üney is with the Institute of Sensors, Signals & Systems of HWU (e-mails: m.üney@hw.ac.uk, d.e.clark@hw.ac.uk).

Simon J. Julier is with the Computer Science Department, University College London, London (e-mail: s.julier@ucl.ac.uk).

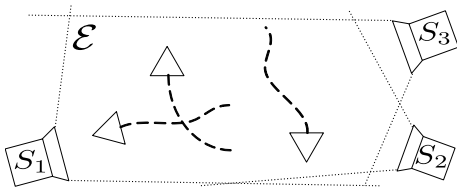


Fig. 1. Illustrative case: Three sensing system nodes S_1, S_2 and S_3 track common targets. Each node measures the target state and exchanges its state estimate with the other nodes.

our distributed multi-object filtering strategy with EMD fusion.

The structure of this paper is as follows: In Section II we present the problem statement and introduce the RFS models for both the multi-target state and sensor observations. An overview of multi-object tracking using RFSs, distributed fusion and the CI approach is given in Section III. Section IV describes our approach for DMMT. We first analytically consider the impact of the EMD on the RFS and derive closed form solutions. We then show how, from this form, a modified version of the PHD filter can be derived. The difficult issue of calculating the weighting parameters is also considered and a scheme based on Rényi divergence is proposed. In Section V, we introduce Monte Carlo methods for realising our EMD fusion approach and present a pseudo-code of the fusion algorithm. The performance of the approach is analysed in a distributed multi-target tracking example in Section VI. Then, we conclude in Section VII.

II. PROBLEM STATEMENT

Consider the scenario illustrated in Figure 1. An environment \mathcal{E} contains a set of targets \mathcal{T} . Neither the number of targets nor the state of each target is known at any given time step k and must be estimated. A set of sensing systems \mathcal{S} are used to monitor \mathcal{E} .

The state of the environment is modelled as the *set* of target states X_k . For an environment with n targets, $X_k = \{x_k^1, \dots, x_k^n\}$, where $x_k^i \in \mathcal{X}$ is the state of the i th target at time step k and \mathcal{X} is the state space. Because both the number of targets and the state of each target is unknown, X_k itself is a *random set*. More formally, a random finite set (RFS) on \mathcal{X} is a measurable mapping $X : \Omega \rightarrow \mathcal{F}(\mathcal{X})$ where $\mathcal{F}(\mathcal{X})$ is the set of all finite subsets of \mathcal{X} and $(\Omega, \sigma(\Omega), \mathcal{P})$ is a probability space. A rigorous construction of probability distributions and densities for random finite sets can be found in, e.g., [13]. In this paper, we assume that the RFS densities we refer to exist.

The transition of a RFS from X_k to X_{k+1} is governed by the transition density $f_{k+1|k}(X_{k+1}|X_k)$. This transition must capture changes to both the cardinality of X , modelling target births and target deaths, as well as the time evolution of surviving targets [8].

Target survival is modelled as a Bernoulli process. The probability that the i th target with state x_k^i will continue to exist at $k+1$ is $p_S(x_k^i)$. If the target exists, its state evolves according to the transition distribution $\pi_{k+1|k}(x_{k+1}^i|x_k^i)$. These processes can be summarised by the following equation:

$$X_{k+1} = \left(\bigcup_{i=1}^{|X_k|} Y_{k+1|k}(x_k^i) \right) \cup \Gamma_{k+1}. \quad (1)$$

Here, $Y_{k+1|k}(x_k^i)$ terms model the evolution of each target that persist to exist. The transition is given by

$$Y_{k+1|k}(x_k^i) = \begin{cases} \{\tilde{x}_k^i\}, & \text{with prob. } p_S(x_k^i) \\ \emptyset, & \text{with prob. } 1 - p_S(x_k^i) \end{cases} \quad (2)$$

where $\tilde{x}_k^i \sim \pi_{k+1|k}(\tilde{x}_k^i|x_k^i)$. $\Gamma_{k+1} = \{b_{k+1}^1, \dots, b_{k+1}^B\}$ is the random set which models target birth. The number of births B is distributed according to $p_b(n)$ and the states b_{k+1}^j are distributed according to $s_b(x)$.

Each sensor outputs a set of detections at time k . A target with state x_k^i is detected with probability $P_D(x_k^i)$. If it is detected, the measurement is characterised by the likelihood $l_k(z_k^i|x_k^i)$. Clutter is modelled using the random finite set C_k , and, hence, the set of measurements is given by

$$Z_k = \left(\bigcup_{i=1}^{|X_k|} Z(x_k^i) \right) \cup C_k. \quad (3)$$

where

$$Z(x_k^i) = \begin{cases} \{z\}, & \text{with prob. } P_D(x_k^i) \\ \emptyset, & \text{with prob. } 1 - P_D(x_k^i) \end{cases} \quad (4)$$

and z is a random vector with density $l_k(z|x_k^i)$. Clutter is modelled using the random finite set $C_k = \{c_1, \dots, c_m\}$ where m and c_i are drawn from the clutter cardinality distribution $p_c(n)$ and localisation distribution $s_c(z)$, respectively.

Given all sensor data, the goal is to construct the posterior density of the multi-target state given by

$$f_k \left(X_k | \{Z_{1:k}^i\}_{i \in \mathcal{S}} \right). \quad (5)$$

One way to achieve this is to pass all the observations from all the sensing systems to a central site where they would be fused together. Although such a centralised scheme is optimal, the need to transmit all observations to a single location could introduce significant communication overheads. Furthermore, such an approach is vulnerable because there is a single point of failure. An alternative is to fuse the data throughout the network. The idea is that each sensing system can be treated as a node in a distributed system. Such nodes collect and process observations locally to create local estimates. These local estimates (rather than raw observations) are periodically broadcast to other nodes where they are fused into that node's state.

Distributed fusion confers many potential advantages. It can be robust to failure — if a single node fails, the other nodes continue to operate and information can be communicated along the remaining of the network. Distributed fusion allows systems to be flexible and scalable — additional nodes, specialised with different processing algorithms and sensing systems can be added and removed upon demand. However, most of these advantages can only be achieved if we impose a *strict locality condition*: each node only knows the identity of its immediate neighbours. As a result, no node needs to know the global topology of the network.

To achieve these goals, we must use algorithms from multi-target tracking and distributed fusion.

III. BACKGROUND

A. Multi-Object Tracking with Random Finite Sets

As explained above, in the Random Finite Set (RFS) paradigm, the multi-target state is represented by the random set X_k . The probability density function for X_k with $|X_k| = n$ is

$$f(X_k) = n!p(n)f(x_k^1, \dots, x_k^n), \quad (6)$$

where $p(n)$ is the probability that the cardinality of the set is n and $f(\cdot)$ is the probability density function for the choice of the state values. The scaling term $n!$ accounts for the fact that $f(\cdot)$ is symmetric with respect to all of its arguments.

Given suitable definitions of the multi-target process and observation models, the multi-target state can be estimated from Bayes rule

$$f_{k|k}(X_k|Z_{1:k}) = \frac{f(X_k|Z_k)f_{k|k-1}(X_k|Z_{1:k-1})}{\int f(Z_k|X_k)f_{k|k-1}(X_k|Z_{1:k-1})\delta X_k} \quad (7)$$

where

$$f_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X_{k-1})f_{k-1|k-1}(X_{k-1}|Z_{1:k-1})\delta X_{k-1} \quad (8)$$

and $\int \cdot \delta X_k$ denotes the set integral defined by

$$\int f(X)\delta X := f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{x_1, \dots, x_n\})dx_1 \dots dx_n. \quad (9)$$

The Bayesian recursion above is practical only for a very small number of targets [8]. Instead, a feasible strategy is to assume that $f_k(X_k|Z_{1:k})$ is a multi-object distribution that can be characterised by its first order moment.

B. Probability Hypothesis Density Filtering

The Probability Hypothesis Density (PHD) is defined as a function which, when integrated over a region \mathcal{R} , gives the expected number of targets in \mathcal{R} [14]. In the RFS paradigm, the first order moment of the distribution, or, the intensity function, $D(x)$ is the PHD [8]. In other words,

$$\int_{\mathcal{R}} D(x)dx = E\{|X \cap \mathcal{R}|\}. \quad (10)$$

Mahler proved that it is possible to construct prediction and update equations directly in terms of the intensity function rather than the full multi-object distribution. For example, in the update stage a *pseudo-likelihood* $L_k(Z_k|x; D_{k|k-1})$ is computed using the observations Z_k and the predicted intensity. The posterior intensity is given by the Bayes-like update

$$D_{k|k}(x|Z_{1:k}) = L_k(Z_k|x; D_{k|k-1})D_{k|k-1}(x|Z_{1:k-1}). \quad (11)$$

Given a certain set of assumptions, computationally cheap approximations can be derived. The two best known of these are the PHD filter [15] and the Cardinalised PHD (CPHD) [16] filter. The PHD filter assumes that the number of targets is Poisson-distributed and its computational costs are linear in

the number of observations. The CPHD uses a general cardinality distribution $p_k(n)$ [16]. Although the computational cost of this algorithm is higher (cubic in the number of measurements), empirical results have shown that it has greater performance than the original algorithm [17].

However, the discussion so far has considered fusion at a single node. We seek methods to fuse data across multiple nodes, and so techniques from distributed data fusion must be considered.

C. Optimal Distributed Fusion

In distributed data fusion (DDF), fusion occurs throughout the entire network rather than at a single, centralised location. Almost all DDF algorithms have been developed for the single target case, under the assumption that techniques such as track-to-track fusion can be used to fuse the different tracks together.

Consider two nodes i and j in the sensor fusion network¹. Each node has received its own set of sensor information and maintains its own posteriors $f(X_k|Z_{1:k}^i)$ and $f(X_k|Z_{1:k}^j)$ respectively. Periodically, node i transmits its posterior to node j . Node j fuses this posterior to compute the joint posterior

$$f(X_k|Z_{1:k}^i, Z_{1:k}^j) = f(X_k|Z_{1:k}^i \cup Z_{1:k}^j). \quad (12)$$

The optimal solution to this problem was developed by Chong, Mori and Chang [18]. They noted that, when fusing the information from different nodes together, $f(X_k|Z_{1:k}^i)$ and $f(X_k|Z_{1:k}^j)$ cannot be assumed to be conditionally independent of one another. There are two reasons for this dependency. The first is that common process noises arise whenever both nodes track the same target. The second is that common observation noise arises after nodes have begun to exchange their local estimates with one another. To correctly model these dependencies, Chong, Mori and Chang proved that

$$f(X_k|Z_{1:k}^i, Z_{1:k}^j) \propto \frac{f(X_k|Z_{1:k}^i)f(X_k|Z_{1:k}^j)}{f(X_k|Z_{1:k}^i \cap Z_{1:k}^j)}. \quad (13)$$

This update rule shows that common information between the nodes must be “divided out”. Various formulations have been derived for a variety of network topologies [5], [6], [19]. However, in almost all cases, $f(X_k|Z_{1:k}^i \cap Z_{1:k}^j)$ can only be computed if some kind of global oracle continuously monitors the entire state of the network. The only case where this does not occur is a tree connected topology, in which a single path exists between any pair of nodes [6]. This makes it possible to compute common information by monitoring the information that flows over edge using so-called “channel filters”. However, tree connected topologies are inherently brittle: The failure of a single node will partition the network. As such, optimal DDF algorithms can only be implemented in highly restricted circumstances and more general formulations are sought.

¹The argument can be readily extended to networks of arbitrary size in which, at any given time, a node is updated only with information from a subset of other nodes in the network.

D. Suboptimal Fusion Using Exponential Mixture Densities

To overcome the problems associated with the optimal update in a distributed setting, Mahler proposed to generalise the Covariance Intersection (CI) to multi-object distributions. Under this generalisation, the familiar product form of Bayes Rule is replaced by taking the geometric mean, or, the exponential mixture of the distributions [20],

$$f_\omega(X_k|Z_{1:k}^i, Z_{1:k}^j) = \frac{f_i(X_k|Z_{1:k}^i)^{(1-\omega)} f_j(X_k|Z_{1:k}^j)^\omega}{\int f_i(X'_k|Z_{1:k}^i)^{(1-\omega)} f_j(X'_k|Z_{1:k}^j)^\omega \delta X'_k} \quad (14)$$

The parameter $\omega \in [0, 1]$ determines the relative weight assigned to each distribution².

The EMD has been analysed traditionally in the context of a single target distribution. In [21], it is shown that the distribution that minimises the weighted sum of its Kullback-Leibler divergence (KLD) with respect to a given set of distributions is an EMD, e.g.,

$$f_\omega = \arg \min_f (1 - \omega)D(f||f_i) + \omega D(f||f_j)$$

where D is the KLD.

It has been shown to automatically prevent *double counting* in arbitrary network topologies [22]. Furthermore, the algorithm can accumulate information (in the sense that the determinant of the covariance matrix of the posterior can be reduced [7] or the peak of the fused distribution can be greater than the peaks of the prior distributions [23]). This generalisation has proved to be extremely valuable for distributed estimation in the single-target case [19], [22]–[24].

Although Mahler proposed the original generalisation of the EMD fusion rule in the context of multi-target tracking using RFSs, no attempt was made to develop fusion rules or algorithms to investigate and realise this generalisation. We now describe how this can be achieved.

IV. DISTRIBUTED FUSION OF PHD FILTERS

In this section, first, we derive explicit formulae for EMDs of RFS distribution families that underlie PHD filters. These formulae enable us to use (14) with posteriors from two PHD [15], CPHD [16] or Bernoulli [25] filters. Then in Section IV-B, we introduce strategies for selecting the EMD weight ω that specifies the fused density.

A. EMDs of RFS Densities

We begin by computing the EMDs of i.i.d. cluster distributions. These are used to fuse posteriors from CPHD filters. Results for Multi-object Poisson and Bernoulli processes follow from this derivation for distributed fusion of the PHD and Bernoulli filters respectively.

²This fusion scheme can readily be generalised to an arbitrary number of sources. For example, if there are n sources, numbered $1, \dots, n$, and if $Z_{1:k}^{1:n}$ is the union of information from all of those sources, then

$$f_\omega(X_k|Z_{1:k}^{1:n}) \propto \prod_{i=1}^n f_i(X_k|Z_{1:k}^i)^{\omega_i}, \quad (15)$$

where $0 \leq \omega_i \leq 1$ and $\sum_{i=1}^n \omega_i = 1$.

Consider two i.i.d. cluster distributions f_i and f_j . These are the posteriors output by two CPHD filters in the network. Omitting the conditioning on the observations for convenience,

$$\begin{aligned} f_i(X) &= n! \cdot p_i(n) \prod_{x \in X} s_i(x), \\ f_j(X) &= n! \cdot p_j(n) \prod_{x \in X} s_j(x). \end{aligned} \quad (16)$$

Proposition 4.1: The EMD of the two i.i.d. cluster distributions in (16), $f_\omega(X)$, is also an i.i.d. cluster process of the form

$$f_\omega(X) = n! \cdot p_\omega(n) \prod_{x \in X} s_\omega(x), \quad (17)$$

where

$$s_\omega(x) = \frac{s_i^{(1-\omega)}(x) s_j^\omega(x)}{Z_\omega(s_i, s_j)}, \quad (18)$$

$$p_\omega(n) = \frac{p_i^{(1-\omega)}(n) p_j^\omega(n) Z_\omega^n(s_i, s_j)}{\sum_{m=0}^{\infty} p_i^{(1-\omega)}(m) p_j^\omega(m) Z_\omega^m(s_i, s_j)}, \quad (19)$$

$$Z_\omega(s_i, s_j) = \int s_i(x)^{(1-\omega)} s_j(x)^\omega dx. \quad (20)$$

Proof: Substituting from (16) into (14), we obtain

$$f_\omega(X) = \frac{1}{K} n! \cdot p_i(n)^{(1-\omega)} p_j(n)^\omega \prod_{x \in X} s_i(x)^{(1-\omega)} s_j(x)^\omega, \quad (21)$$

where K is the set integral of the numerator over all finite subsets of the state space. Considering the set integral (9)

$$\begin{aligned} K &= \sum_{m=0}^{\infty} p_i(m)^{(1-\omega)} p_j(m)^\omega \left(\int s_i(x')^{(1-\omega)} s_j(x')^\omega dx' \right)^m \\ &= \sum_{m=0}^{\infty} p_i(m)^{(1-\omega)} p_j(m)^\omega Z_\omega^m(s_i, s_j) \end{aligned} \quad (22)$$

where $Z_\omega(s_i, s_j)$ is given by (20).

After multiplying the numerator and denominator of (21) with $Z_\omega(s_i, s_j)^n$, we obtain

$$f_\omega(X) = n! \cdot \frac{p_i(n)^{(1-\omega)} p_j(n)^\omega Z_\omega^n(s_i, s_j)}{K} \prod_{x \in X} s_\omega(x), \quad (23)$$

where $s_\omega(x)$ is a probability density over the single object state space and given by (18). This expression is in the form of an i.i.d. cluster where the cardinality distribution is identified as (19) after substituting from (22) into (23). ■

Proposition 4.1 lets us fuse the posterior distributions propagated by CPHD filters. The localisation density of the fused distribution (18) is the EMD of the input localisation densities s_i and s_j . The fused cardinality density, on the other hand, is the scaled product of fractional powers of input cardinalities and the scale factor (20) of the fused localisation density raised to power n .

Given this form, the PHD of the update is directly given by

$$D_\omega(x) = \mu_\omega s_\omega(x), \quad (24)$$

where

$$\mu_\omega = \sum_{n=1}^{\infty} n p_\omega(n) \quad (25)$$

is the expected number of targets.

EMD fusion of two PHD filters follows from Proposition 4.1 after substituting Poisson cardinality densities in (16):

Corollary 4.2: Consider two multi-object Poisson distributions given by (16) where $p_i(n)$ and $p_j(n)$ are Poisson densities with parameters μ_i and μ_j , respectively. The corresponding EMD is a multi-object Poisson with the cardinality distribution parameter

$$\mu_\omega = \mu_i^{(1-\omega)} \mu_j^\omega \cdot Z_\omega(s_i, s_j) \quad (26)$$

and localisation distribution given by (18).

Proof: Proposition 4.1 holds for multi-object Poisson distributions as they constitute a subclass of i.i.d. clusters with Poisson cardinalities. Therefore, it suffices to show that the EMD cardinality distribution is Poisson. Consider the EMD cardinality distribution given by (19) after substituting Poisson densities:

$$\begin{aligned} p_\omega(n) &= \frac{1}{K} \left(\frac{\mu_i^n \exp(-\mu_i)}{n!} \right)^{(1-\omega)} \left(\frac{\mu_j^n \exp(-\mu_j)}{n!} \right)^\omega Z_\omega(s_i, s_j)^n \\ &= \frac{1}{K} \frac{(\mu_i^{(1-\omega)} \mu_j^\omega Z_\omega(s_i, s_j))^n}{n!} \exp(-\mu_i(1-\omega) - \mu_j\omega) \end{aligned} \quad (27)$$

where the denominator K is found by

$$\begin{aligned} K &= \sum_{m=0}^{\infty} \left(\frac{\mu_i^m \exp(-\mu_i)}{m!} \right)^{(1-\omega)} \left(\frac{\mu_j^m \exp(-\mu_j)}{m!} \right)^\omega Z_\omega(s_i, s_j)^m \\ &= \exp(-\mu_i(1-\omega) - \mu_j\omega) \sum_{m=0}^{\infty} \frac{(\mu_i^{(1-\omega)} \mu_j^\omega Z_\omega(s_i, s_j))^m}{m!} \\ &= \exp(-\mu_i(1-\omega) - \mu_j\omega) \exp\left(\mu_i^{(1-\omega)} \mu_j^\omega Z_\omega(s_i, s_j)\right) \end{aligned} \quad (28)$$

After substituting from (28) into (27), we obtain

$$p_\omega(n) = \frac{(\mu_i^{(1-\omega)} \mu_j^\omega Z_\omega(s_i, s_j))^n \exp(-\mu_i^{(1-\omega)} \mu_j^\omega Z_\omega(s_i, s_j))}{n!}$$

which is a Poisson distribution with parameter given by (26). ■

For EMD fusion of Bernoulli filters, we use Proposition 4.1 with Bernoulli cardinalities:

Corollary 4.3: Consider two Bernoulli RFS distributions given by (16) where $p_i(n)$ and $p_j(n)$ are Bernoulli densities with parameters α_i and α_j respectively, i.e.,

$$p_i(n) = \begin{cases} 1 - \alpha_i, & n = 0, \\ \alpha_i, & n = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (29)$$

and $p_j(n)$ has similar values with parameter α_j .

The EMD is a Bernoulli RFS distribution with the cardinality parameter

$$\alpha_\omega = \frac{\alpha_i^{(1-\omega)} \alpha_j^\omega Z_\omega(s_i, s_j)}{(1 - \alpha_i)^{(1-\omega)} (1 - \alpha_j)^\omega + \alpha_i^{(1-\omega)} \alpha_j^\omega Z_\omega(s_i, s_j)} \quad (30)$$

and localisation distribution given by (18).

Proof: Proposition 4.1 holds for Bernoulli RFS distributions as they constitute a subclass of i.i.d. clusters with Bernoulli cardinalities. Therefore, it suffices to show that the EMD cardinality distribution is Bernoulli. After substituting Bernoulli densities in (19) and evaluating $p_\omega(n)$ for $n = 0, 1, 2, \dots$, we obtain

$$p_\omega(n) = \begin{cases} \frac{1}{K} (1 - \alpha_i)^{(1-\omega)} (1 - \alpha_j)^\omega, & n = 0, \\ \frac{1}{K} \alpha_i^{(1-\omega)} \alpha_j^\omega Z_\omega(s_i, s_j), & n = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

where

$$K = (1 - \alpha_i)^{(1-\omega)} (1 - \alpha_j)^\omega + \alpha_i^{(1-\omega)} \alpha_j^\omega Z_\omega(s_i, s_j) \quad \blacksquare$$

B. Choosing the EMD Weighting Parameter

Unlike Bayes rule, the EMD fusion rule is *active* in the sense that the mixture parameter ω must be specified. This parameter controls the relative weighting on f_i and f_j . Suppose that $J(\omega)$ is a cost function. The goal is to choose ω such that

$$\omega^* = \arg \min_{\omega \in [0,1]} J(\omega). \quad (32)$$

Motivated by the original derivation of CI, possible choices for $J(\omega)$ include the determinant or the trace of the covariance of f_ω [26]. However, when the distribution is multi-modal, the covariance is not necessarily a good representation of uncertainty. Another possibility is to consider the Shannon Entropy of f_ω [27]. However, the entropy can contain local minima, making the optimisation in (32) difficult to solve [12]. Therefore, we seek alternative measures that are easy to solve for but still convey potentially useful information.

In his probabilistic analysis of CI, Hurley [27] proposed the criteria that the Kullback–Leibler divergence (KLD) of f_ω from f_i and f_j should be the same. Specifically, (32) can be selected as

$$J(\omega) = (\mathcal{D}(f_\omega || f_i) - \mathcal{D}(f_\omega || f_j))^2. \quad (33)$$

Although Hurley's arguments strictly apply to just discrete distributions, Dabak has generalised then to continuous distributions [28]. Furthermore he demonstrated that $\mathcal{D}(f_\omega || f_i)$ is a non-decreasing function of ω . As a result, (33) possesses a unique minimum which greatly simplifies the optimisation problem. However, the information theoretic justification for using the divergence measure as a cost function is unclear. Hurley argued his choice on the ground that the resulting distribution is related to the *Chernoff Information*. However, this is associated with binary classification problems, and its relevance to information fusion is unclear. An alternative interpretation is to consider this as an example of the Principle of Minimum Discrimination Information Theorem [29].

Although the KLD is a useful metric, the Rényi Divergence (RD) has been found to be more useful in sensor management problems [30]. The RD generalises the KLD through the introduction of a free parameter α which can be used to emphasise particular aspects of the differences between the distributions which are of interest such as its tails. For $\alpha \rightarrow 1$

the Rényi divergence converges to the KLD. For $\alpha = 0.5$, it equals the Hellinger affinity and the weight selection criterion becomes the equality of Hellinger distances.

Using the formula for Rényi divergence between two iid cluster processes (see, e.g., [30]), it can easily be shown that

$$\begin{aligned} R_\alpha(f_\omega(X)||f_i(X)) &= \frac{1}{\alpha-1} \log \sum_{n=0}^{\infty} p_\omega(n)^{1-\alpha} p_i(n)^\alpha \left[\int_{\mathcal{X}} s_\omega(x)^{(1-\alpha)} s_i(x)^\alpha dx \right]^n \\ &= \frac{1}{\alpha-1} \log \sum_{n=0}^{\infty} p_\omega^\alpha(n) p_i^{(1-\alpha)}(n) \left[\frac{Z_{\alpha\omega}(s_i, s_j)}{(Z_\omega(s_i, s_j))^\alpha} \right]^n \end{aligned} \quad (34)$$

where $Z_{\alpha\omega}(s_i, s_j)$ and $Z_\omega(s_i, s_j)$ are obtained using (20).

Following similar steps, the divergence of the EMD with respect to the second distribution $f_j(X)$ can be found by

$$\begin{aligned} R_\alpha(f_\omega(X)||f_j(X)) &= \frac{1}{\alpha-1} \log \sum_{n=0}^{\infty} p_\omega^\alpha(n) p_j^{(1-\alpha)}(n) \left[\frac{Z_{\alpha(1-\omega)}(s_j, s_i)}{(Z_\omega(s_i, s_j))^\alpha} \right]^n. \end{aligned} \quad (35)$$

The numerical computation of these quantities is discussed in Section V-D.

Further discussion on the use of different choices of $J(\omega)$ and different α values can be found in [12]. The results suggest that the divergence measures are easier to implement than other classes of cost measures and have minimal impact on the overall performance of the system.

V. IMPLEMENTATION OF THE EMD FUSION ALGORITHM

There are two main challenges when implementing the EMD fusion rule. The first is that the EMD rarely admits a closed form solution. For example, if $s_i(x)$ and $s_j(x)$ in (16) are Gaussian mixture models (GMMs), the weighted geometric mean computed in (18) will not, in general, be another GMM. Although a Newton series expansion can be used to approximate it as a GMM, the series can become numerically unstable unless an extremely large number of components are used [31]. Therefore, robust methods for computation are required. The second issue is that the update needs to incorporate the effects of the ω selection strategy outlined in Section IV-B. Since this is an optimisation process, the updated distributions and divergence values must be repeatedly calculated for different values of ω . Therefore, efficient calculation schemes are required.

When a node processes information collected locally, a conventional SMC implementation of the CPHD filter is used. Because of its efficiency with spawning new targets, we use the Adaptive Birth Process (ABP) proposed by Ristić, Clark, Vo and Vo [32]. To fuse data from another sensing system, we must be able to compute (17), (18) and (20) for a range of values of ω . However, this cannot be carried out directly because each node has its own particle filter with its own support. Therefore, we use clustering techniques to create continuous approximations of the distributions. These distributions are then sampled from to compute the EMDs using different particle support.

We now discuss each of these steps in turn.

A. Fusion of Local Information

Each node maintains a local i.i.d. cluster distribution. For the i th node, this can be written as $\{p_{k|k}^i(n), \{\zeta_{k|k}^{(m)}, x_i^{(m)}, l^{(m)}\}_{m=1:M_i}^i\}$. $p_{k|k}^i(n)$ is the cardinality distribution. The three remaining terms store the information associated with each of the M_i particles used to represent the localisation distribution. The first component, $\zeta_{k|k}^{(m)}$, is the weight associated with the m th particle $x_i^{(m)}$ which is a point generated from the localisation distribution. Given these two components, the localisation distribution is computed from

$$\hat{s}_{k|k}(x|Z_{1:k}) = \sum_{m=1}^{M_k} \zeta_{k|k}^{(m)} \delta(x - x_{k|k}^{(m)}). \quad (36)$$

Given the average number of targets,

$$\mu_{k|k} = \sum_{n=0}^{n_{max}} np(n), \quad (37)$$

where n_{max} is the length of the storage for $p(n)$, the PHD is

$$\hat{D}_{k|k}(x|Z_{1:k}) = \mu_{k|k} \hat{s}_{k|k}(x|Z_{1:k}). \quad (38)$$

The final term, $l^{(m)}$, is a particle label assigned by the ABP which initialises potentially new targets by creating a set of particles for each measurement. We maintain the measurement-particle associations by labeling particles according to the measurement that originated them [32]. Because the label identifies the measurement which originated that particle, this information can be used when fusing the distributions from different nodes together.

B. Continuous Approximation of SMC CPHD for Distributed Fusion

A node i cannot directly fuse the intensity from node j . Because each node uses its own SMC representation, each node will have its own set of particles. Neither the support — nor the number of particles — are guaranteed to be the same in each. Therefore, we seek a continuous approximation which can be used. Such approximation problems are solved through Kernel Density Estimation (KDE) methods [33], in which the estimated density is a sum of kernel functions shifted to particle points. Fraley demonstrated how model-based clustering methods can be used for density estimation [34]. However, Fraley's approach is not robust to outliers which can cause a high degree of uncertainty in the cardinality distribution and can lead to many mixture components. Instead, we exploit the labels present in the ABP to create the clusters. We associate each cluster $l \in \{l_1, \dots, l_L\}$ with a set of parameters \mathbf{C}_l and use the density estimate given by

$$\hat{s}(x|Z_{1:k}) = \frac{1}{M} \sum_{m=1}^M K(x, x^{(m)}; \mathbf{C}_{l^{(m)}}), \quad (39)$$

where $K(x, x^{(m)}; \mathbf{C}_{l^{(m)}})$ is a Gaussian with mean $x^{(m)}$ and covariance $\mathbf{C}_{l^{(m)}}$.

Next, we describe the computation of \mathbf{C}_l for cluster l :

In order to find the kernel parameters \mathbf{C}_l for the members of the cluster l , i.e., $\{x^{(m')} | l^{(m')} = l\}$, we first find a transform that diagonalises the empirical covariance of these points in the transformed domain. Then, the problem of finding the kernel parameters in multiple-dimensions reduces to independent single dimensional problems.

This transform is given by the inverse square root of the empirical covariance matrix $\mathbf{\Sigma}_l$ of cluster l . We transform all $x^{(m')} \in \{x^{(m')} | l^{(m')} = l\}$ using

$$\begin{aligned} y^{(m')} &= \mathbf{W}_l x^{(m')}, \\ \mathbf{W}_l &= \mathbf{\Sigma}_l^{-1/2}. \end{aligned}$$

Given that the covariance of $y^{(m')}$ is diagonal, the d -dimensional Gaussian kernel in the transformed domain simplifies to

$$K(y, y^{(m')}) = \prod_{k=1}^d \frac{1}{\sqrt{2\pi}h_k} \exp\left(-\frac{1}{2} \frac{(y_k - y_k^{(m')})^2}{h_k^2}\right)$$

where d is the dimensionality of the state space and h_k s are the bandwidth (BW) parameters of the 1-D Gaussian kernels [33].

The BW h_k for each dimension can be found using one of the well established methods in the literature [35]. In particular, we use the following rule-of-thumb (RUT) [33]:

$$h_k = \sigma_k \left(\frac{4}{3N}\right)^{1/5}$$

where σ_k is the empirical standard deviation of $y_k^{(j)}$ s and N is the number of these points. The reason for this choice is its simplicity and low computational complexity compared to other methods such as *least squares cross-validation* [35].

Therefore, the covariance matrix that specify the kernels in (39) for the members of the cluster l is found as

$$\begin{aligned} \mathbf{C}_l &= \mathbf{T}_l \mathbf{\Lambda}_l \mathbf{T}_l^T \\ \mathbf{T}_l &= \mathbf{W}_l^{-1} \\ \mathbf{\Lambda}_l &= \text{diag}(h_1^2, h_2^2, \dots, h_d^2) \end{aligned}$$

C. Construction of the EMDs

In this Section, we consider the EMD in Proposition 4.1 and introduce Monte Carlo methods to construct the multi-object EMD for any $\omega \in [0, 1]$. We first introduce a sampling procedure to generate particles representing the localisation density $s_\omega(x)$ given by (18). We then find the cardinality distribution (19) after estimating the scale factor Z_ω given by (20).

1) *Sampling from the EMD Localisation Distribution:* We consider sampling from the fused localisation density (18) using equally weighted sets of particles $\{x_i^{(m_i)}\}_{m_i=1:M_i}$ and $\{x_j^{(m_j)}\}_{m_j=1:M_j}$ together with KDE parameters $\{C_{l_i}\}_{l_i \in \mathcal{L}_i}$ and $\{C_{l_j}\}_{l_j \in \mathcal{L}_j}$ representing $s_i(x)$ and $s_j(x)$ respectively.

The consistency properties of the EMDs [23] motivate the use of non-degenerate mixtures of $s_i(x)$ and $s_j(x)$ as proposal densities for importance sampling (IS). The reason for this is that, in our experience, these mixtures tend to have heavier

tails [36] than that of $s_\omega(x)$. The union of the input particle sets, i.e.,

$$P_U \triangleq \{x_i^{(m_i)}\}_{m_i=1:M_i} \cup \{x_j^{(m_j)}\}_{m_j=1:M_j} \quad (40)$$

is constituted of $M_\omega = M_i + M_j$ samples from the mixture density

$$s_p(x) = \frac{M_i s_i(x) + M_j s_j(x)}{M_i + M_j}. \quad (41)$$

Therefore, P_U given by (40) is a convenient particle set to represent $s_\omega(x)$ in which case the IS weights for $x^{(m')} \in P_U$ are given by

$$\zeta^{(m')} \propto \frac{s_i^{(1-\omega)}(x^{(m')}) s_j^\omega(x^{(m')})}{M_i s_i(x^{(m')}) + M_j s_j(x^{(m')})}. \quad (42)$$

After resampling $\{x^{(m')}, x^{(m')}\}_{m'=1:M_i+M_j}$ for $M_i + M_j$ times, one obtains samples approximately generated from $s_\omega(x)$. In order to compute the IS weights (42), however, evaluation of both $s_i(x)$ and $s_j(x)$ at all points of P_U is necessary. For estimating these values, we use the KDE parameters $\{C_{l_i}\}_{l_i \in \mathcal{L}_i}$ and $\{C_{l_j}\}_{l_j \in \mathcal{L}_j}$ within (39) and obtain the KDEs $\hat{s}_i(x)$ and $\hat{s}_j(x)$ respectively. Then, we evaluate $\hat{s}_i(x)$ and $\hat{s}_j(x)$ at P_U . Hence, feasible estimates of $\zeta^{(m')}$ s are computed by substituting these quantities in (42):

$$\hat{\zeta}^{(m')} \propto \frac{\hat{s}_i^{(1-\omega)}(x^{(m')}) \hat{s}_j^\omega(x^{(m')})}{M_i \hat{s}_i(x^{(m')}) + M_j \hat{s}_j(x^{(m')})}. \quad (43)$$

After resampling, $\{x^{(m)}, x^{(m)}\}$, we obtain equally weighted samples representing $s_\omega(x)$.

2) *Construction of the EMD Cardinality Distribution:* In order to compute the fused cardinality distribution given by (19), one needs to estimate $Z_\omega(s_i, s_j)$ given by (20). Using the proposal density $s_p(x)$ given in (41), the IS estimate [36] of this quantity is given by

$$\tilde{Z}_\omega(s_i, s_j) \triangleq \sum_{x \in P_U} \frac{s_i^{(1-\omega)}(x) s_j^\omega(x)}{M_i s_i(x) + M_j s_j(x)} \quad (44)$$

where P_U is the union of the input particles sets (40).

We substitute the KDEs $\hat{s}_i(x)$ and $\hat{s}_j(x)$ in (44) to achieve computational feasibility and obtain

$$\hat{Z}_\omega(s_i, s_j) \triangleq \sum_{x \in P_U} \frac{\hat{s}_i^{(1-\omega)}(x) \hat{s}_j^\omega(x)}{M_i \hat{s}_i(x) + M_j \hat{s}_j(x)} \quad (45)$$

After estimating the scale factor, $p_\omega(n)$ can be constructed by substituting $p_i(n)$, $p_j(n)$ and $\hat{Z}_\omega(s_i, s_j)$ in (19) for $n = 0, 1, \dots, n_{\max}$ where $n_{\max} + 1$ is the length of the storage array.

D. Computation of the Rényi divergences

We consider the estimation of the Rényi divergences $R_\alpha(f_\omega(X) || f_i(X))$ and $R_\alpha(f_\omega(X) || f_j(X))$ given by (34) and (35), respectively, to use within (33). Given α and ω , we first construct the EMD cardinality $p_\omega(n)$ as described in Section V-C2. In order to do that, one evaluates the KDEs $\hat{s}_i(x)$ and $\hat{s}_j(x)$ at P_U only once. Then, $Z_\omega(s_i, s_j)$, $Z_{\alpha\omega}(s_j, s_i)$ and $Z_{\alpha(1-\omega)}(s_j, s_i)$ are estimated with these evaluations using (45) with the only difference being the

TABLE I
INITIAL TARGET STATES AND TRACK INFORMATION.

Init. loc. (m.)	Init. vel. (m/s)	time of birth/death (s.)
$[1333.3, 6866.7]^T$	$[0.0, -136.7]^T$	0/70
$[-4000.0, 4000.0]^T$	$[106.7, -106.7]^T$	10/85
$[-6135.3, 1594.1]^T$	$[137.6, 12.9]^T$	25/95
$[-4000.0, 5000.0]^T$	$[120.0, -40.0]^T$	35/110
$[4000.0, 4000.0]^T$	$[-106.7, -106.7]^T$	50/120

value of the subscript parameter in the left hand side. Finally, we substitute these quantities into (34) and (35) and (33). Note that the KDE evaluations are the most resource demanding procedures required for fusion. Nevertheless all MC computations described in Sections V-C–V-D use the same set of evaluations. Therefore, the computational overhead of divergence evaluations in EMD fusion is negligible.

E. The MC Multi-object EMD fusion algorithm

We use the MC procedures developed in Section IV. A pseudo-code of the proposed fusion algorithm is given in Algorithm 1. The first inputs of the algorithm are the local and the incoming particle representations of the i.i.d. cluster posteriors. Then, the α parameter of the Renyi divergence is entered which is used for calculating the cost $J(\omega)$ in (33). Finally, an increment value Δ is input for finding the best EMD weight ω^* by exhaustive search³.

First, the KDE parameters of the particle sets are found. Then, the sample set P_U from the proposal distribution is constructed, and, KDEs of the input localisation densities are evaluated at the particles in this set. Once the KDEs are evaluated at P_U , the Rényi divergences of the EMD with respect to the inputs and the cost in (33) is computed while ω is varied with Δ increments starting from $\omega = 0$. After the costs are found over the grid specified by Δ , the best EMD weight ω^* is found. In the following step, IS weights of the proposal samples are computed for ω^* .

The output of the algorithm is a set of particles representing the fused localisation density $s_{\omega^*}(x)$ and the fused cardinality distribution array $p_{\omega^*}(n)$. The most computationally demanding step in the algorithm is the evaluation of the KDEs. Since we need to perform this step only once before the for loop, the computational cost of the exhaustive search remains negligible.

VI. EXAMPLES

In this Section, we test the performance of the EMD fusion algorithm using the distributed tracking scenario shown in Fig. 2. Four fixed range-bearing sensors (S_1 , S_2 , S_3 and S_4) observe the environment \mathcal{E} in which 5 targets appear and disappear over time. The state of each target is defined by its position $[x, y]$ and velocity $[\dot{x}, \dot{y}]$.

The tracks are obtained by evolving target states $[x, y, \dot{x}, \dot{y}]^T$ in accordance with a linear constant velocity motion model and (slight) additive zero mean process noise. The initial states

³In Section IV-B, we discussed that when equality of divergences are used for selecting the EMD weight ω , one can safely use convex optimisation methods as well. In this paper, however, we perform an exhaustive search over a grid in $[0, 1]$ values of which are defined by the increment Δ .

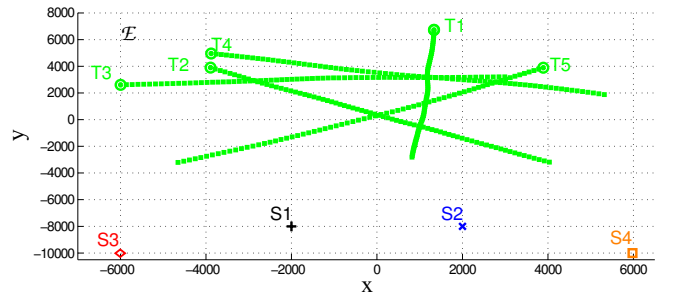


Fig. 2. Target tracks and sensor locations for the example scenario: Sensors 1, 2, 3 and 4 are shown with black '+', blue 'x', red 'o' and orange '□', respectively. The target tracks shown by green lines are initiated at the locations shown by circles. The birth and death times are given in Table I.

together with times of birth and death are given in Table I. The observation model for each sensor is the same — the standard deviations in range and bearing are 5m and 2° respectively. The probability of detection in each sensor is independent of the probability of detection at all other sensors and is $P_D = 0.90$. The number of clutter reports in each scan is Poisson distributed with $\lambda = 12$. Each clutter report is sampled uniformly over \mathcal{E} .

A. EMD Fusion of a Sensor Pair

In this example we consider fusion of the sensor pair S_1 and S_2 . Platform S_2 regularly transmits its posterior to S_1 at every time step. Three algorithms were tested:

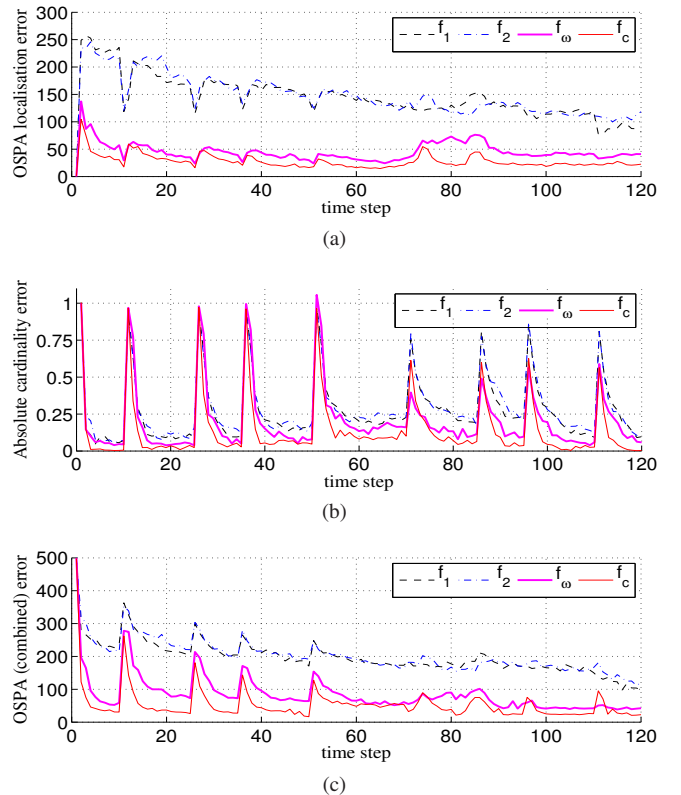


Fig. 3. Average performances of filter 1 (black-dashed line), filter 2 (blue-dashed line), the EMD fusion (magenta line) and the centralised filter (red line): (a) OSPA localisation error, (b) Absolute cardinality error, (c) (combined) OSPA error.

Algorithm 1 Multi-object EMD Fusion Algorithm for Monte Carlo realisations at local sensor i .

Inputs: $(p_i, \{x_i^{(m_i)}, l_i^{(m_i)}\}_{m_i=1:M_i})$ of the local sensor i and $(p_j, \{x_j^{(m_j)}, l_j^{(m_j)}\}_{m_j=1:M_j})$ from sensor j ,
 α parameter of the Renyi divergence,
 Δ for the increments of ω .

Find $\{C_{l_i}\}_{l_i \in \mathcal{L}_i}$ and $\{C_{l_j}\}_{l_j \in \mathcal{L}_j}$ as described in Section V-B,

Find P_U given by (40),

Find the KDE evaluations $S_i \triangleq \{\hat{s}_i(x) | x \in P_U\}$ and $S_j \triangleq \{\hat{s}_j(x) | x \in P_U\}_{m_i=1:M_i}$ using the KDE parameters $\{C_{l_i}\}_{l_i \in \mathcal{L}_i}$ and $\{C_{l_j}\}_{l_j \in \mathcal{L}_j}$ in (39).

for $\omega = 0, \Delta, \dots, 1$ **do**

Estimate $Z_\omega, Z_{\alpha\omega}$ and $Z_{\alpha(1-\omega)}$ using S_i and S_j in (45) \triangleright Note that S_i and S_j are evaluated before entering the loop.

Find p_ω using the estimated Z_ω in (19)

Find $\mathcal{R}_\alpha(f_\omega, f_i)$ using p_ω , the estimated Z_ω and $Z_{\alpha\omega}$ in (34)

Find $\mathcal{R}_\alpha(f_\omega, f_j)$ using p_ω , the estimated Z_ω and $Z_{\alpha(1-\omega)}$ in (35)

Find $J(\omega)$ using $\mathcal{R}_\alpha(f_\omega, f_i)$ and $\mathcal{R}_\alpha(f_\omega, f_j)$ in (33)

end for

Find $\omega^* = \arg \min_{\omega \in \{0, \Delta, \dots, 1\}} J(\omega)$,

Find IS weights $\zeta^{(m')}$ for $\omega = \omega^*$ and for each $x^{(m')} \in P_U$ using S_i and S_j in (43), $\triangleright p_{\omega^*}$ is already found in the **for** loop.

Store the labels $\mathcal{L} = \mathcal{L}_i \cup \mathcal{L}_j$ and the KDE parameters $\mathcal{C} \triangleq \{C_{l_i}\}_{l_i \in \mathcal{L}_i} \cup \{C_{l_j}\}_{l_j \in \mathcal{L}_j}$

Return: $(p_{\omega^*}, \{x^{(m')}, \zeta^{(m')}, l^{(m')}\}_{m'=1:M_i+M_j})$ and \mathcal{C}

- 1) **No fusion.** Each node operates independently of the other node and use the SMC CPHD filter with ABP (Section V). We use 1500 particles per persistent target and 300 particles are generated per observation for modeling target births. The survival probability $P_S = 0.98$ and the new born target intensity is selected as $0.9e-3$.
- 2) **Centralised fusion.** The measurements are sent to a central site and fused together (using the iterated corrector approximation [37]).
- 3) **Distributed fusion.** The EMD fusion scheme was used. We select the Rényi divergence parameter $\alpha = 0.5$ and the search increment $\Delta = 0.01$. With these conditions, we seek an EMD which is equi-distant to both posteriors in the Hellinger distance sense.

Fig. 3 shows the performance of the different algorithms assessed using the OSPA metric. We use the cut-off parameter $c = 500m$, and the exponent parameter $d = 1$ which allows us to use OSPA localisation also as a distance metric. Graphs are presented for the OSPA localisation error, the absolute error between the expected value of the cardinality distribution and the true number of targets, and the combined OSPA distance averaged over 150 Monte Carlo runs. The results demonstrate that fusing data from the different sensing systems can greatly reduce both the localisation and the cardinality errors. Not surprisingly, the centralised scheme, which optimally fuses all estimates together, produces the smallest errors. However, the distributed fusion algorithm also shows significant improvements in performance.

The computational times for the algorithms vary with the number of particles. Using nonoptimized MATLAB code on an 8-core 2.7 GHz. laptop with 8 GBs of memory, one iteration of the CPHD filter takes a maximum of 2.467s, and, a minimum of 0.675s is reached when only a single target exists. The maximum and minimum average times for fusion are

4.322×10^1 s and 4.555s, respectively. KDE evaluations during these computations take 4.192×10^1 s and 4.296s, respectively. The for loop evaluating the information cost over a 100 point grid for ω takes maximum and minimum values of 0.834s

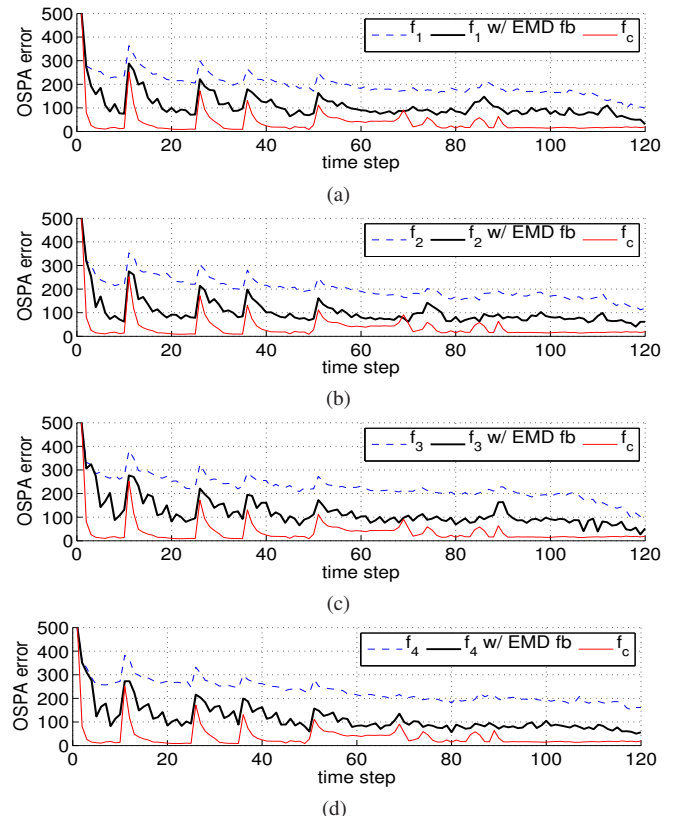


Fig. 4. (a)–(d) Average (combined) OSPA error for platforms 1 – 4, respectively, using feedback from EMD fusion. The upper bounds are the performances for (myopic) filtering of local observations and the lower bound is the performance of the centralised filter.

and 0.225s. The steps of the fusion algorithm including the KDE evaluations can be parallelised. If these computations are carried out using, for example, graphical processing units (GPUs), substantial speed-ups can be obtained [38].

B. EMD Fusion of Multiple Sensors

In this example, we consider fusion of multiple sensors assuming that the communication constraints do not allow for transmission of frequent updates among sensor platforms and each platform can receive at most one posterior from one of its neighbours at a given time. The platforms which receive posteriors from their neighbours first employ EMD fusion and then replace the local posterior maintained by the local filter with the fusion output – to be updated in the next time step using the local observations and the CPHD filter prediction/update. Note that the fact that EMDs prevent double-counting of information allows this feedback scheme to be used without affecting the optimality of the prediction and update stage of filtering in the following time step.

We use a communication regime which involves repeating a pattern for the transmissions starting at time step 2. At this step, the transmitter-receiver pairs are selected as $\{(S_3, S_1), (S_4, S_2)\}$. Upon receiving the posterior from a neighbour, each of S_1 and S_2 employs EMD fusion and replace their local posterior with the fusion output. At time step 3, the transmission pattern is $\{(S_1, S_2), (S_2, S_1)\}$. At timestep 4, the selected pairs are $\{(S_1, S_3), (S_2, S_4)\}$. This communication pattern means that all nodes communicate with all other nodes.

We present the performance gain provided by EMD fusion using (combined) OSPA error with respect to the ground truth for all platforms 1 – 4 in Fig.4(a)-(d), respectively. The results are averaged over 150 Monte Carlo runs. The upper bounds (blue dashed lines) are obtained by averaging (combined) OSPA error for solely filtering local observations. The lower bound (solid red line) is the OSPA performance of the centralised filter. Local filters receiving feedback from EMD fusion (solid black lines) perform significantly better than myopic filtering and fairly close to the centralised result under an infrequent communication regime.

VII. CONCLUSION

In this work we have investigated and developed a novel strategy for robust, distributed multi-sensor multi-object tracking. In particular, we have considered distributed fusion of Probabilistic Hypothesis Density filters through a generalisation of the Covariance Intersection fusion rule to multi-object EMDs. We have introduced practical algorithms using Monte Carlo methods and evaluated the performance of these algorithms in challenging scenarios through statistical simulations. EMD fusion significantly improves upon the performances of myopic multi-object filters through local communications among nodes over an unknown, dynamic global network topology. Future work will involve the investigation of these algorithms for jointly fusion and sensor registration.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Branko Ristić for his help and support throughout this research.

REFERENCES

- [1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic Press, 1988.
- [2] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control*, vol. 24 no. 6, 1979.
- [3] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*, ser. Artech House radar library. Artech House, 1999.
- [4] G. Castañón and L. Finn, "Multi-Target Tracklet Stitching Through Network Flows," in *Proceedings of the 2011 IEEE Aerospace Conference*, Big Sky, MT, USA, 5–12 March 2011, pp. 1–7.
- [5] M. E. Liggins II, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 95–107, 1997.
- [6] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor fusion networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.
- [7] J. K. Uhlmann, "Dynamic Map Building and Localization for Autonomous Vehicles," Ph.D. dissertation, University of Oxford, 1995.
- [8] R. P. S. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
- [9] R. Mahler, "Optimal/robust distributed data fusion: a unified approach," *SPIE Defense and Security Symposium*, 2000.
- [10] D. Clark, S. Julier, R. Mahler, and B. Ristić, "Robust multi-object sensor fusion with unknown correlations." SSPD 2010, September 2010.
- [11] M. Üney, S. Julier, D. Clark, and B. Ristić, "Monte carlo realisation of a distributed multi-object fusion algorithm." SSPD 2010, September 2010.
- [12] M. Üney, D. Clark, and S. Julier, "Information Measures in Distributed Multitarget Tracking," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, July 2011, pp. 1–8.
- [13] B. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for Multi-target Filtering with Random Finite Sets." *IEEE Trans. Aerospace Elec. Systems*, vol. 41, No.4, pp. 1224–1245, 2005.
- [14] R. Mahler, "A theoretical foundation for the stein-winter "probability hypothesis density(phd)" multitarget tracking approach," in *Proceedings of the National Symposium on Sensor and Data Fusion*, vol. 1, 2000, (Unclassified) Sandia National Laboratories.
- [15] —, "Multi-target bayes filtering via first-order multi-target moments," *IEEE Transactions on Aerospace and Electronic Sysmtes*, vol. 39, no. 4, pp. 1152–1178, October 2003.
- [16] —, "PHD Filters of Higher Order in Target Number," *IEEE Transactions on Aerospace and Electronic Sysmtes*, vol. 43, no. 4, pp. 1523–1543, October 2007.
- [17] B. T. Vo, B.-N. Vo, and A. Cantoni, "Analytic Implementations of Probability Hypothesis Density Filters." *IEEE Transactions on Signal Processing*, Vol 55 No 7 Part 2, pp. 3553–3567, 2007.
- [18] C.-Y. Chong, S.Mori, and K.-C. Chang, "Distributed multitarget multi-sensor tracking," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Artech House, 1990, ch. 8.
- [19] L.-L. Ong, T. Bailey, and H. Durrant-Whyte, "Decentralised Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks," in *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, 30th June – 3rd July 2008, pp. 1–8.
- [20] R. Mahler, "Optimal/robust distributed data fusion: a unified approach," in *Proceedings of the SPIE Defense and Security Symposium 2000*. SPIE Defense and Security Symposium, 2000.
- [21] T. Heskes, "Selecting weighting factors in logarithmic opinion pools," in *Advances in Neural Information Processing Systems*. The MIT Press, 1998, pp. 266–272.
- [22] S. Julier, "Fusion without independence (keynote abstract)," in *Proceedings of the IET Seminar on Tracking and Data Fusion: Algorithms and Applications*, 15–16 April 2008, pp. 1–5.
- [23] S. J. Julier, T. Bailey, and J. K. Uhlmann, "Using exponential mixture models for suboptimal distributed data fusion," in *Proc. of the 2006 IEEE Nonlinear Stat. Signal Proc. Workshop*. Cambridge, UK: NNSPW'06, September 2006, pp. 160–163.
- [24] S. Julier, "An Empirical Study into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models," *IEEE Int. Conf. on Information Fusion*, 2006.

- [25] B.-T. Vo, D. Clark, B.-N. Vo, and B. Ristic, "Bernoulli forward-backward smoothing for joint target detection and tracking," *Signal Processing, IEEE Transactions on*, vol. 59, no. 9, pp. 4473–4477, Sept. 2011.
- [26] S. Julier and J. Uhlman, *General Decentralized Data Fusion with Covariance Intersection (CI)*. CRC Press, 2001, ch. 12.
- [27] M. B. Hurley, "An Information-Theoretic Justification for Covariance Intersection and Its Generalization," in *Proceedings of the 2002 FUSION Conference*, vol. 1. FUSION 2002, July 2002, pp. 505–511.
- [28] A. Dabak, "A Geometry for Detection Theory," Ph.D. dissertation, Rice University, Houston, TX, USA, 1992.
- [29] S. Kullback and M. A. Khairat, "A Note on Minimum Discrimination Information," *Annals of Mathematical Statistics*, vol. 37, pp. 279–280, 1966.
- [30] B. Ristić, B.-N. Vo, and D. Clark, "A Note on the Reward Function for PHD Filters with Sensor Control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 72–80, April 2011.
- [31] N. Mariam, "Conservative Non-Gaussian Data Fusion for Decentralized Networks," Master's thesis, University of Sydney, August 2007, last accessed 19th December, 2012. [Online]. Available: <http://db.acfr.usyd.edu.au/content.php/292.html?publicationid=910>
- [32] B. Ristić, D. Clark, B.-N. Vo, and B.-T. Vo, "Adaptive Target Birth Intensity for PHD and CPHD Filters," *IEEE Trans Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1656–1668, April 2012.
- [33] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [34] C. Fraley and A. W. Raftery, "Model based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, June 2002.
- [35] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 401–407, March 1996.
- [36] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer, 2004.
- [37] R. Mahler, "Approximate multisensor cphd and phd filters," in *Information Fusion (FUSION), 2010 13th Conference on*. IEEE, 2010, pp. 1–8.
- [38] B. Srinivasan, Q. Hu, and R. Duraiswami, "Gpuml: Graphical processors for speeding up kernel machines," in *Proc. of the Workshop on High Performance Analytics-Algorithms, Implementations, and Applications*, 2010.