DISTRIBUTED MULTI-SENSOR CPHD FILTER USING PAIRWISE GOSSIPING

Jun Ye Yu, Mark Coates, and Michael Rabbat

McGill University
Montreal, Canada

ABSTRACT

We present a distributed *cardinalized probability hypothesis density* (CPHD) filter for multi-sensor multi-target tracking. Each sensor runs a single-sensor CPHD filter to compute the *probability hypothesis density* (PHD) function and cardinality distribution using only its own measurements and then fuses the local results by gossiping with neighboring sensors. Existing schemes that fuse local results using the Kullback-Leibler average are adversely affected if some sensors do not detect a target. The proposed fusion strategy, based on the arithmetic mean instead of the geometric mean, aims to be more robust to missed detections. We also show via simulations that the performance of the proposed algorithm can be significantly improved, with a small additional communication overhead, by having sensors exchange measurements locally.

Index Terms— Multi-target tracking, distributed tracking, probability hypothesis density, pairwise gossip

1. INTRODUCTION

Multi-target tracking is an essential building block in many applications. A network of sensors collects noisy measurements to locate one or more targets. If multiple sensors are present, the measurements from all sensors should be processed for improved tracking performance. While transmitting all data to a central unit is a possible solution, a distributed approach is much preferred because there is no communication bottleneck and the network is more resistant to sensor failure.

In this paper, we present a distributed multi-sensor *cardinalized probability hypothesis density* (CPHD) filter in which sensors use randomized pairwise gossip [1] to exchange and fuse their PHD functions and cardinality distributions. In particular, sensors represent the PHD function using a *Gaussian mixture model* (GMM), and fusion is accomplished via the arithmetic mean, rather than the geometric mean used in [2]. We argue that using the arithmetic mean leads to more robustness when not all sensors detect a target present, and simulation results illustrate that the proposed approach adapts more quickly to changes in the number of targets present.

1.1. Related work

Representing the target states as a *random finite set* (RFS) [3, 4] is an attractive solution for multi-target tracking because the two unknowns of interest, the number of targets and their states, can be captured in one single random variable. This also provides a single integrated framework for data association and tracking. The PHD [5] and CPHD [6] filters have been extensively studied, and various implementations, including using Gaussian mixtures [7, 8] and sequential Monte Carlo [9], have been proposed. The PHD filter only estimates the distribution of target states and assumes that the target cardinality follows a Poisson distribution. The CPHD filter also estimates the target cardinality distribution and achieves better performance at the cost of higher computational overhead. While both filters are designed for single-sensor tracking, extensions to the multi-sensor scenario [10, 11, 12, 12] have also been proposed. An alternative approach is the multi-Bernoulli filter [3, 13] which uses Bernoulli RFSs that are characterized by a target state distribution and a target existence probability.

In the domain of distributed multi-target tracking, the consensus CPHD filter proposed in [2] has each sensor compute its own estimate, and then sensors fuse their results with those of their neighbors. The fusion strategy involves computation of the weighted Kullback-Leibler average [14, 15] of local PHD functions. A similar approach has been applied to obtain a distributed multi-Bernoulli filter [16]. Datta Gupta et al. [17] have proposed a distributed multi-sensor CPHD filter in which sensors sequentially process and update their local PHD and cardinality distributions. The first sensor processes its own measurements and passes the data to the second sensor which then integrates its own measurements. The chain continues and the output from the last sensor is considered the final estimate and disseminated backwards through the chain. This approach mimics the actual update step of the general multi-sensor CPHD filter [12], but requires maintaining a Hamiltonian path.

1.2. Motivation

After two neighboring nodes communicate, the consensus CPHD filter computes the weighted Kullback-Leibler average of their PHD functions, $D^a(x)$ and $D^b(x)$, which is given by the weighted geometric mean [14]:

$$D^{a,b}(x) = \frac{[D_a(x)]^\omega [D_b(x)]^{1-\omega}}{\int [D_a(x)]^\omega [D_b(x)]^{1-\omega} dx} \qquad (1)$$

where $\omega \in [0, 1]$ is a user-defined weight parameter. For pair-wise averaging between two sensors with the same probability of detection and clutter rate, one would take $\omega = 1/2$.

Let x_t denote the state of one target. If sensor a does not detect a target and sensor b does, then we have $D^a(x_t) \approx 0$ and $D^b(x_t) > 0$. After fusing the two PHD functions, $D^{a,b}(x_t) \approx 0$ and the target is lost even though the target was detected by at least one sensor.

1.3. Contribution

This work proposes a distributed multi-sensor CPHD filter with a gossip-based fusion strategy to address the issue illustrated above. Rather than taking the geometric mean we use the arithmetic mean of the PHD functions. We argue that this leads to more robust fusion when not all sensors detect a target. The experiments reported in Sec. 3 illustrate that this leads to faster overall detection of a new target by the system, and hence better tracking accuracy.

2. DISTRIBUTED GOSSIP CPHD FILTER

2.1. Algorithm overview

We can divide a distributed multi-sensor tracking algorithm into three stages: prediction, update and fusion. At each time step, each sensor runs the prediction and update stages locally using only its own measurements. In the fusion stage, sensors communicate with each other to reach consensus on their estimates.

In the proposed algorithm, each sensor runs a single-sensor CPHD filter for the prediction and update stages. In the fusion stage, sensors fuse their PHD functions and cardinality distributions over a number of gossip iterations. We describe the fusion strategy in detail in Section 2.2. Note that, at each gossip iteration, only two sensors communicate with each other. In contrast, in one consensus iteration [2], each sensor communicates sequentially with all of its neighbors.

2.2. Fusion of PHD functions and cardinality distributions

Consider a network of S sensors, and assume that they have identical detection probabilities and clutter rates. Let N_s denote the neighbors of sensor s with $s' \in N_s$ if sensor s can receive data from sensor s'. By definition, $s \in N_s$. Let $D_{k|k}^s(x)$ and $p_{k|k}^s(n)$ denote the estimated PHD function and cardinality distribution of sensor s at time k after running the prediction and update stages. Since each sensor only has access to its own measurements, the estimates at different sensors most likely differ ($D_{k|k}^s(x) \neq D_{k|k}^{s'}(x), p_{k|k}^s(n) \neq p_{k|k}^{s'}(n), \forall s \neq s'$). The objective is to fuse all sensors' local PHD functions and cardinality distributions, and our algorithm achieves this by having sensors gossip with their neighbors over a number of iterations.

Let us focus on one gossip iteration and assume sensors a and b are gossiping with the sensors' PHD functions being

$$D^s(x) = \sum_{i=1}^{L_s} w_i^s \mathcal{N}(x; m_i^s, P_i^s), \quad s \in \{a, b\},$$

where L_s is the number of mixture components in sensors s's PHD function, m_i^s are the component means, and P_i^s are the covariance matrices. The weights w_i^s are normalized so that $\sum_{i=1}^{L_s} w_i^s = 1$.

When sensors a and b gossip, the two individual GMMs are first concatenated:

$$D^{a,b}(x) = \sum_{i=1}^{L_a} w_i^a \mathcal{N}(x; m_i^a, P_i^a) + \sum_{j=1}^{L_b} w_j^b \mathcal{N}(x; m_j^b, P_j^b).$$

We then apply a merging/pruning algorithm to the concatenated GMM (see Algorithm 1). Note that the same merging/pruning algorithm is also used in GMM-based implementations of the PHD/CPHD filter [7, 8]. The algorithm requires three threshold parameters: T, U, and J_{max}. Mixture components with weights below T are truncated. Two components are only merged if their Mahalanobis distance is below the threshold U, and finally, at most the J_{max} mixture components with the highest weights are retained. These thresholds prevent the size of the GMM representation (in particular, the number of mixture components) from growing without bound. In a distributed algorithm, this ensure the communication overhead of gossiping remains bounded.

The resulting GMM after merging is the fused PHD function. Note that, in the gossip update, even if only one of the sensors detects

---

Algorithm 1 Gaussian mixture merging/pruning [7]

Input: J Gaussian mixture components $\{w_i, m_i, P_i\}_{i=1}^J$, truncation threshold T, merging threshold U, maximum number of components J_{max}
1: Set $l = 0$
2: Set $I = \{i = 1, 2, ...J | w_i > T\}$
3: **while** $I \neq \varnothing$ **do**
4: $\quad l = l + 1$
5: $\quad j = \arg\max_{i \in I} w_i$
6: $\quad L = \{i \in I | (m_i - m_j)^T (P_i)^{-1} (m_i - m_j) \leq U\}$
7: $\quad \tilde{w}_l = \sum_{i \in L} w_i$
8: $\quad \tilde{m}_l = \frac{1}{\tilde{w}_l} \sum_{i \in L} w_i x_i$
9: $\quad \tilde{P}_l = \frac{1}{\tilde{w}_l} \sum_{i \in L} w_i \left(P_i + (\tilde{m}_l - m_i)(\tilde{m}_l - m_i)^T \right)$
10: $\quad I = I \backslash L$
11: **if** $l > J_{max}$ **then**
12: $\quad$ sort $\{\tilde{w}_i\}_{i=1}^l$: $\tilde{w}_1 \geq \tilde{w}_2 \geq ...\tilde{w}_l$
13: $\quad$ retain only $\{\tilde{w}_i, \tilde{m}_i, \tilde{P}_i\}_{i=1}^{J_{max}}$
14: $\tilde{w}_i = \tilde{w}_i / \sum_i \tilde{w}_i$
Output: $\{\tilde{w}_i, \tilde{m}_i, \tilde{P}_i\}$

---

a target ($D^a(x_t) > 0$ and $D^b(x_t) \approx 0$), the target is not necessarily lost in the fused PHD function.

For fusing the cardinality distributions, we simply average the two sensors' local distributions: $p^{a,b}(n) = \frac{p^a(n) + p^b(n)}{2}$.

At each time step, we run a fixed number of gossip iterations to drive the local PHD functions and cardinality distributions sufficiently close to each other. While formally defining convergence and computing convergence rate are beyond the scope of this paper, simulations show that as the number of gossip iterations grows, the difference between different sensors' PHD functions and cardinality distributions tends to zero.

In terms of communication overhead, the gossip approach requires $2G_{max}$ transmissions where G_{max} is the number of gossip iterations per time step. The consensus approach requires $2SE(|N_s|)C_{max}$ transmissions where C_{max} is the number of consensus iterations and each sensor communicates with $E(|N_s|)$ neighbors per iteration, on average. Each transmission contains the posterior estimates from a single sensor. Conservatively bounding $G_{max} \leq S^2/2$ and $C_{max} \geq 1$, the gossip approach requires less overhead than the consensus approach when $E(|N_s|) > 0.5S$.

We note that using the (unweighted) arithmetic mean may be problematic if not all sensors have equal properties (e.g., if some have lower probability of detection or higher clutter rates than others). In such a case, it may be more appropriate to take a weighted average of the PHD function and cardinality distribution. We leave a more thorough study of this case for future work.

2.3. Incorporating neighboring sensors' measurements

We have so far made the restriction that sensors can only access their own measurements to compute the local PHD function (predict and update). However, if they have also access to their neighbors' measurements (which could be done by having each sensor broadcast its measurements before the prediction stage), then this information can be used, using the multi-sensor CPHD filter update equations [18] to derive a more accurate estimate of the PHD function before running the gossip iterations. We illustrate this idea and show the resulting performance gain via simulation in Section 3.3.
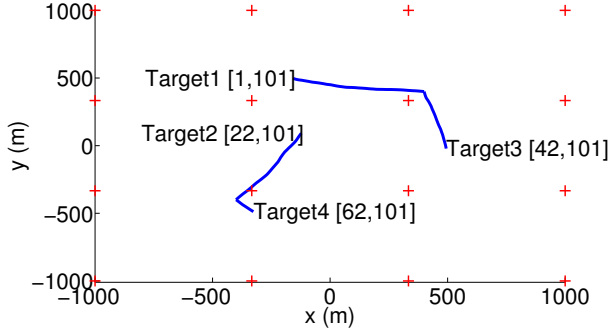
Fig. 1. Simulation target trajectories. The number in brackets indicate the time interval during which the target is present. The labels indicate targets' final position, and '+'s indicate sensor positions.

3. PERFORMANCE EVALUATION

Next we evaluate the performance of the proposed algorithm. We describe the simulation setup in Section 3.1. In Section 3.2, all sensors only use their own measurements to compute the local estimates prior to fusion. In Section 3.3, sensors exchange measurements with their neighbors prior to computing the local estimates.

3.1. Simulation setup

We consider a network of 16 sensors spread over an area of $2000 \times 2000 \ m^2$. Initially, there is only one target in the tracking area. Three additional targets gradually appear over time. Fig. 1 illustrates the target trajectories and sensor positions.

The target state is defined as $x(k) = [x_1(k), x_2(k), \dot{x}_1(k), \dot{x}_2(k)]$ where $x_1(k), x_2(k)$ denote the target's position at time k and $\dot{x}_1(k), \dot{x}_2(k)$ denote its velocity. The target dynamic follows a nearly constant linear velocity model:

$$ x(k+1) = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + w(k), $$

where $T_s = 1s$ is the sampling interval and $w(k)$ is the zero-mean Gaussian process noise with covariance matrix Q and noise intensity $\sigma_w = 0.25$:

$$ Q = \sigma_w^2 \begin{bmatrix} T_s^3/3 & 0 & T_s^2/2 & 0 \\ 0 & T_s^3/3 & 0 & T_s^2/2 \\ T_s^2/2 & 0 & T_s & 0 \\ 0 & T_s^2/2 & 0 & T_s \end{bmatrix}. $$

We also assume a linear measurement model:

$$ z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(k) + v(k) $$

where $v(k)$ is the zero-mean Gaussian measurement noise with covariance $R = \text{diag}([100, 100])$.

At each iteration, every sensor makes a set of measurements. Each measurement corresponds either to a target or to clutter, and each target can account for up to one measurement at each sensor. Each sensor detects each target independently and with probability P_d, and clutter is modeled as a Poisson process with parameter $\lambda_c = 10$ measurements per sensor and uniform spatial distribution.

We compare three algorithms in our simulation: the *general multi-sensor CPHD* (GMCPHD) [12, 18], the consensus CPHD [2], and the proposed gossip CPHD filters. For all three algorithms, we model the target birth process as a two-component GMM with means $[400, 400, 0, 0]$ and $[-400, -400, 0, 0]$. Both components have weight 0.1 and covariance matrix equal to $\text{diag}([100, 100, 25, 25])$. The target survival probability is $P_s = 0.99$. The cardinality distribution has finite support with maximal cardinality equal to 20. For the merging algorithm, we set $T = 10^{-5}$, $U = 4$ and $J_{max} = 100$.

For the two distributed algorithms, sensors i and j are connected if $c_{ij} < P_c$ where $c_{ij} \in U(0, 1)$ and $P_c \in [0, 1]$ is the connectivity probability. We assume bidirectional communication ($i \in N_j \Leftrightarrow j \in N_i$) as required for gossiping. We set $G_{max} = 100$ for the gossip CPHD filter and $C_{max} = 1$ for the consensus CPHD filter.

We use the *optimal subpattern assignment* (OSPA) error [19] as our performance metric with cardinality penalty factor $c = 25$ and power $p = 1$. For the two distributed algorithms, we compute the OSPA for each sensor at every time step and report the average. Unless otherwise specified, all results are averaged over 200 Monte-Carlo trials. Each trial has the same target trajectories, but different realizations of clutter and measurement noise. The adjacency matrix for the network changes for each trial. For the gossip CPHD filter, the gossiping sensors at different time steps also differ for each trial.

3.2. Local measurements only

Fig. 2(a) compares the three algorithms' performance with respect to target detection probability. The gossip CPHD filter consistently outperforms the consensus CPHD filter with the gap widening as we lower P_d. At high detection probability, the performance of the gossip CPHD filter approaches that of the centralized GMCPHD filter.

Next, we consider the impact of network connectivity, which directly translates into the number of neighbors each sensor may have. As shown in Fig. 2(b), the proposed algorithm's performance is very consistent over a wide range of P_c. This is not surprising because, as long as the network remains connected, gossiping allows the sensors to reach a consensus, albeit at a slower rate when network connectivity is low. On the other hand, the performance of the consensus CPHD filter degrades slightly as we increase P_c. Since evaluating Eq. (1) requires some approximations [20, 2] when the PHD functions are modeled as GMMs, as the number of neighbors increases, we conjecture that the error from the approximation also accumulates. If no fusion occurs ($P_c = 0$) and each sensor can only rely on its own estimates, the average OSPA is much higher.

We compare the estimated target cardinality over time, averaged over all sensors and all trials, in Fig. 2(c). All three algorithms are able to estimate the target cardinality with high accuracy. Whenever a new target appears, the proposed gossip CPHD filter is able to adapt to the new cardinality faster than the consensus CPHD filter does.

In lieu of a formal proof of convergence of the proposed gossip fusion rule, we illustrate the idea by computing the total discrepancy between the PHD functions of all sensors as a function of the number of gossip iterations. We adopt a distance metric proposed by Surajit [21] to compute the discrepancy between two GMMs:

$$ l(D^i(x), D^j(x)) = \int (D^i(x) - D^j(x))^2 dx, \qquad (2) $$

and define the total discrepancy as

$$ L = \sum_{i=1}^{S} \sum_{j=i+1}^{S} l(D^i(x), D^j(x)). \qquad (3) $$

(a) OSPA with respect to detection probability   (b) OSPA with respect to connectivity probability   (c) Estimated vs true cardinality, $P_d = 0.7$, $P_c = 0.5$
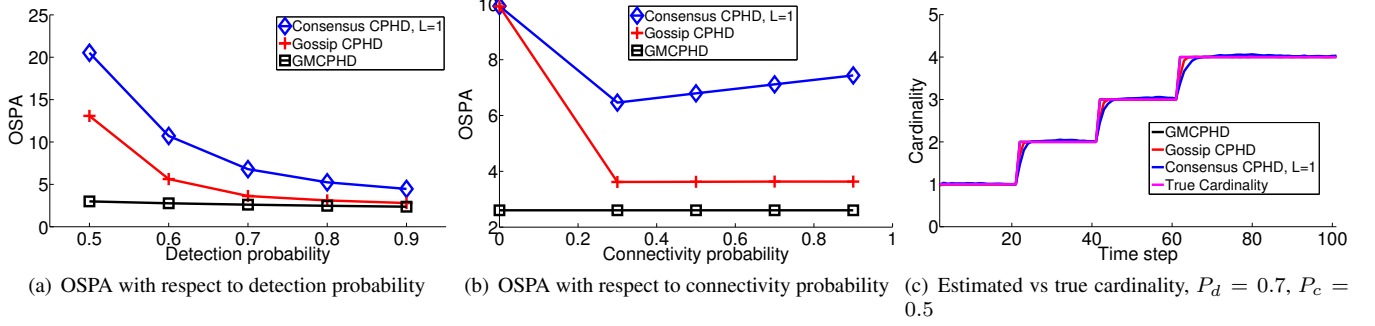
Fig. 2. Algorithm performance comparison. In (a), we vary the probability of target detection. In (b), we vary the connectivity probability. In (c), we compare the average estimated cardinality of all 16 sensors over time. All results are averaged over 200 random trials.
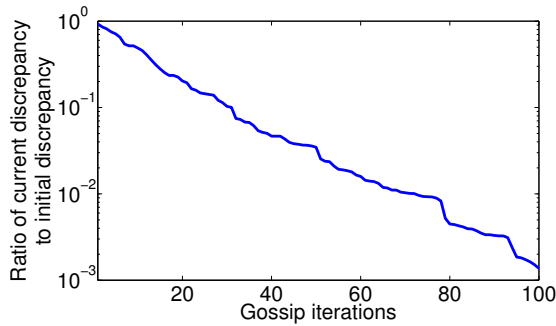

Fig. 3. Discrepancy ratio vs. gossip iterations, $P_d = 0.7$, $P_c = 0.6$.

Fig. 4. Algorithm performance with respect to number of neighbors each sensor can exchange measurement data with. At 0, each sensor only has access to its own measurements. $P_d = 0.7$, $P_c = 0.5$.

In Fig. 3, we compute the total discrepancy at time step 80. The vertical axis corresponds to the ratio of the discrepancy at the current gossip iteration to the discrepancy prior to gossiping. The figure suggests that, for the type of random network topology simulated, the total discrepancy decreases roughly exponentially with the number of pairwise gossip iterations.

3.3. Access to neighboring sensors' measurements

Next we consider the extension proposed in Section 2.3 and allow sensors to exchange measurement data with their neighbors. At the beginning of each time step, all sensors broadcast their measurements to their neighbors. Each sensor then runs a GMCPHD filter [18] locally to process its own measurements as well as those of its neighbors. We assume that that each sensor receives all measurement data from its neighbor(s). As shown in Fig. 4, as we increase the number of neighbors each sensor can share measurement data with, the performance improves.

Another approach similar to our extension is to disseminate each sensor's measurements to the entire network. Once sensors have the measurements from all nodes in the network, they can derive the same estimates without gossiping. It is thus worth comparing the communication overhead in both approaches.

Assume each sensor has an average of 14 measurements (one from each target plus ten clutter measurements). Disseminating one sensor's measurements to the entire network requires $\mathcal{O}(\sqrt{S/\log S})$ transmissions [22]. Thus, for $S = 16$ sensors, we need to transmit approximately 1100 scalars.

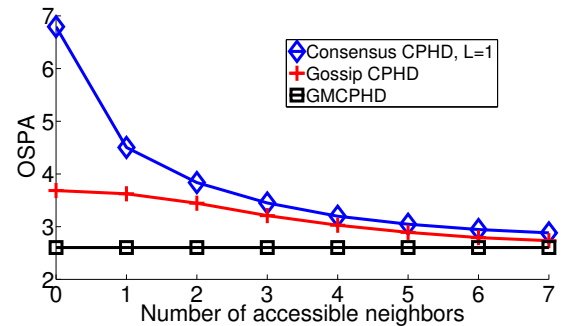Next, assume each sensor's PHD contains 8 components (up to

2 per target). Each component can be represented by 15 scalars. The cardinality distribution is represented by 21 scalars. For 100 pairwise gossip iterations, the total overhead is approximately 28000 scalars.

Although gossiping has a much higher overhead, it may still be desirable or necessary in certain circumstances. For example, if communication is unreliable and not all measurements are successfully disseminated to all other nodes, then the local PHD and cardinality estimates will gradually diverge unless some additional synchronization mechanism, such as gossip or consensus, is used.

4. CONCLUSION

In this paper, we present a distributed CPHD filter for multi-target tracking. Sensors compute the PHD function and cardinality distribution using their own measurements and fuse their results by gossiping with their neighbors. Simulations show that the proposed algorithm outperforms existing techniques based on Kullback-Leibler average. Furthermore, by allowing sensors to exchange measurement data with their neighbors, the proposed algorithm's performance improves significantly. Our future work includes establishing a formal proof of convergence for the proposed gossip updates, bounding the error between gossip CPHD fusion and GMCPHD updates, and generalizing the gossip fusion rule to handle sensors with different measurement characteristics (P_d and clutter rate).

5. REFERENCES

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

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

[3] R. P. S. Mahler, *Statistical Multisource-multitarget Information Fusion*, Artech House, 2007.

[4] I. R. Goodman, R. P. Mahler, and H. T. Nguyen, *Mathematics of data fusion*, Springer Science & Business Media, 2013.

[5] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.

[6] R. P. S. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, 2007.

[7] B.-N Vo and W.K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, 2006.

[8] B.-T. Vo and B.-N. B.-N. Vu. Antonio, "Analytic ipmlementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, 2007.

[9] 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, 2005.

[10] R. P. S. Mahler, "The multisensor PHD filter: II. Erroneous solution via Poisson magic," in *SPIE Int. Conf. Sig. Proc., Sensor Fusion, Target Recognition. XVIII*, 2009.

[11] R. P. S. Mahler, "Approximate multisensor CPHD and PHD filters," in *Int. Conf. Inf. Fusion*, 2010.

[12] S. Nannuru, M. Coates, M. Rabbat, and S. Blouin, "General solution and approximate implementation of the multisensor multitarget CPHD filter," in *Int. Conf. Acoust. Speech and Signal Process.*, 2015.

[13] B.-T. Vo, B.-N. Vo, and C. Antonio, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, 2009.

[14] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[15] G. Battistelli, L. Chisci, A. Farina, and B.-N. Vo, "Average Kullback-Leibler divergence for random finite sets," in *Int. Conf. Inf. Fusion*, 2015.

[16] M. B. Guldogan, "Consensus Bernoulli filter for distributed detection and tracking using multi-static Doppler shifts," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 672–676, 2014.

[17] S. Datta Gupta, S. Nannuru, M. Coates, and M. Rabbat, "A distributed general multi-sensor cardinalized probability hypothesis density (CPHD) filter for sensor networks," in *Proc. SPIE Signal Process. Sensor/Inform. Fusion and Target Recognition XXIII*, 2015.

[18] S. Nannuru, S. Blouin, M. Coates, and M. Rabbat, "Multisensor CPHD filter," preprint, available online at http://arxiv.org/abs/1504.06342, 2015.

[19] 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, 2008.

[20] S. J. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *Int. Conf. Inf. Fusion*, 2006.

[21] R. Surajit, *Distance-based model-selection with application to the analysis of gene expression data*, Ph.D. thesis, The Pennsylvania State University, 2003.

[22] M. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.