

# INITIALIZATION OF MULTI-BERNOULLI RANDOM-FINITE-SETS OVER A SENSOR TREE

Joseph Lee and K. Yao

CENS & Electrical Engineering Department, University of California, Los Angeles, CA, USA

## ABSTRACT

We study a framework of multi-target state initialization employed over a tree-like sensor network. Such network provides graceful aggregation of target state statistics by exploiting limited bandwidth, and makes it possible to approximate the target states based on particle filtering defined in the context of Random-Finite-Set (RFS) theory. This contrasts with conventional multi-target filtering approach based on RFS over a star topology. In the tree structure, the root sensor node launches the initialization and passes messages downward to its leaf sensors. Each leaf sensor node collects upward messages from its subsidiary nodes and updates target state statistics. We implement the initialization process through the Sequential Monte-Carlo (SMC) method, which requires the sampling and re-sampling steps different from that in a non-RFS type initialization and aggregation.

**Index Terms**— Initialization, Multi-target state, Random-finite sets, Multi-Bernoulli, Sensor network

## 1. INTRODUCTION

We consider a tree-like sensor network [1] that aggregates information of multi-target states. The network consists of a root node and its sub-tree of leaf nodes. Messages that carry multi-target state statistics traverse from root downward to the bottom of the tree, and reflect upward back to the root (Fig. 1). In this paper we use sensor and node interchangeably.

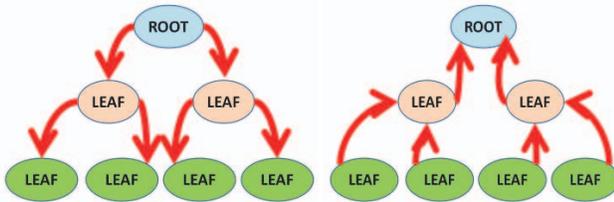


Fig. 1. Downward message pass (left) and upward message pass (right).

We adopt the model of multi-targets along the line of multi-Bernoulli Random-Finite-Set (RFS) theory [2]. RFS provides a comprehensive Bayesian framework when one tackles a point pattern with a time-varying number of time-varying random variables. As in non-RFS Bayesian filtering, initialization of RFS posterior functions is an indispensable step. The result of initialization not only allows each sensor to perform target tracking locally, but also enables the fusion center to aggregate information over a sensor network and perform target tracking centrally.

Previous treatments of RFS Bayesian fusion have mostly focused on star-like sensor network and ignored the deficit in latency. Owing to packet collisions, a star structure may introduce latency as worse as a function linear in the number of sensors. The latency may be improved to logarithm of the number of sensor in a tree structure.

Fusion of posterior RFS functions implemented in Sequential Monte-Carlo (SMC) method requires sampling and re-sampling steps different from its counterpart in non-RFS Bayesian filtering. Surprisingly, explicit equations for RFS-type multi-sensor posterior fusion have been nearly rare in previous literature [3]. To the best of our knowledge, many researchers only hint the use of likelihood function factorization, which makes implementation of multi-target initialization incomplete to base on SMC method. In this paper, we address with these challenges in initialization of RFS Bayesian filtering over a sensor tree.

### 1.1. Multi-Bernoulli RFS

For our application of multiple two dimensional targets tracking, we consider a four-tuple  $[p_1, p_2, p'_1, p'_2]^T$  that indicates position and velocity (both defined in  $\mathbb{R}^2$ ). A random target state  $X$  takes set-value in  $\mathcal{F}(\mathcal{X})$ , the space of finite subsets of  $\mathcal{X} \subseteq \mathbb{R}^4$ . Assume  $M$  possible tracks exist in the multi-target model. Consider a union of  $M$  independent RFSs  $X = \cup_{i=1}^M X^{(i)}$ , where each  $X^{(i)}$  obeys the probability density in (1).

$$\pi(X^{(i)}) = \begin{cases} 1 - r^{(i)}, & X^{(i)} = \emptyset \\ r^{(i)} p(x^{(i)}), & X^{(i)} = \{x^{(i)}\} \end{cases} \quad (1)$$

The density function  $\pi(X^{(i)})$  characterizes a Bernoulli RFS  $X^{(i)} \sim \{r^{(i)}, p^{(i)}(\cdot)\}$  with probability  $1 - r^{(i)}$  of being empty and probability  $r^{(i)}$  of being a singleton  $X^{(i)} \sim p^{(i)}(\cdot)$ . We say  $X \sim \{r^{(i)}, p^{(i)}(\cdot)\}_{i=1}^M$  is a Multi-Bernoulli RFS with probability density in (2) [4].

$$\pi(X) = \begin{cases} \pi(\emptyset) \equiv \prod_{i=1}^M (1 - r^{(i)}), & X = \emptyset \\ \pi(\emptyset) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq M} \prod_{j=1}^n \frac{r^{(i_j)} p^{(i_j)}(x_j)}{1 - r^{(i_j)}}, & X = \{x_1, \dots, x_n\} \end{cases} \quad (2)$$

Let  $X_n$  denote one realization of the multi-target state set to be estimated by sensor  $n$  with cardinality  $M(n)$ ,  $X_n = \{x_{n,1}, \dots, x_{n,M(n)}\} \in \mathcal{F}(\mathcal{X})$ , and  $Z_n$  denote one realization of the multi-target measurement set observed by sensor  $n$  with cardinality  $N(n)$ ,  $Z_n = \{z_{n,1}, \dots, z_{n,N(n)}\} \in \mathcal{F}(\mathcal{Z})$ , where  $\mathcal{Z}$  represents the random measurement set defined in a sensor-specific domain.

As the spatial correlation among sensors may not be available, each sensor may not infer cross-sensor posterior conditioned on its own measurements. Hence, one faces the need to fuse posterior density  $\pi(X|Z)$ . Then a new challenge arises in RFS Bayesian filtering. The choice of importance density for generating particles remains elusive in previous RFS literature. Some authors proposed to use transition density for sampling/re-sampling in time-evolving RFS filtering [3]. However, there is no transition density necessary or available in the problem we study herein, which requires a novel approach to developing importance density for initialization.

## 1.2. Multi-Sensor Data Fusion

To fuse data from two sensors (indexed by 1 and 2) at time 0, we exploit the respective likelihood functions  $g_0^{(1)}$  and  $g_0^{(2)}$ , the prior  $\pi_0$  and a measure  $\mu(\delta X)$  to derive the posterior conditioned on the joint data set as shown in (3).

$$\begin{aligned} \pi_0(X_0 | Z_0^{(1)}, Z_0^{(2)}) &= \frac{g_0(Z_0^{(1)}, Z_0^{(2)} | X_0) \pi_0(X_0)}{\int g_0(Z_0^{(1)}, Z_0^{(2)} | X) \pi_0(X) \delta X} \\ &= \frac{g_0^{(1)}(Z_0^{(1)} | X_0) g_0^{(2)}(Z_0^{(2)} | X_0) \pi_0(X_0)}{\int g_0^{(1)}(Z_0^{(1)} | X) g_0^{(2)}(Z_0^{(2)} | X) \pi_0(X) \delta X} \end{aligned} \quad (3)$$

## 2. MESSAGE PASS

In this section we show how initial multi-target state statistics evolve over a sensor tree. We note that the SMC approach to realizing particles and weights in RFS Bayesian fusion requires a more comprehensive treatment. Let  $X$  denote the multi-target state to be initialized. All nodes have been installed identical prior  $\pi_0(\cdot) = \{r_0^{(i)}, p_0^{(i)}(\cdot)\}_{i=1}^{M_0}$  with particles and weights  $\{w_0^{(i,j)}, x_0^{(i,j)}\}_{j=1}^{L_0^{(i)}}$  to approximate  $p_0^{(i)}$ , as shown in (4).

$$p_0^{(i)}(x) \approx \sum_{j=1}^{L_0^{(i)}} w_0^{(i,j)} \delta_{x_0^{(i,j)}}(x) \quad (4)$$

### 2.1. SMC Initialization Per Node

After the root node launches network-wide initialization, each node  $n$  computes legacy tracks  $\{r_{L,n}^{(i)}, p_{L,n}^{(i)}\}_{i=1}^{M_0}$  and measurement-corrected tracks  $\{r_{U,n}(z), p_{U,n}(z)\}_{z \in Z_n}$  by evaluating the multi-Bernoulli approximation of the updated multi-target posterior  $\pi_n(\cdot | Z_n)$  for  $\{r_{L,n}^{(i)}, p_{L,n}^{(i)}\}_{i=1}^{M_0} \cup \{r_{U,n}(z), p_{U,n}(z)\}_{z \in Z_n}$  as follows (with similar notations given in [4]).

$$q_{L,n}^{(i)} = \sum_{j=1}^{L_0^{(i)}} w_0^{(i,j)} p_{D,n}(x_0^{(i,j)}), \quad (5)$$

$$w_{L,n}^{(i,j)} = w_0^{(i,j)} (1 - p_{D,n}(x_0^{(i,j)})), \quad (6)$$

$$\tilde{w}_{L,n}^{(i,j)} = \frac{w_{L,n}^{(i,j)}}{\sum_{j=1}^{L_0^{(i)}} w_{L,n}^{(i,j)}}, \quad (7)$$

$$q_{U,n}^{(i)}(z) = \sum_{j=1}^{L_0^{(i)}} w_0^{(i,j)} \psi_{n,z}(x_0^{(i,j)}), \quad (8)$$

$$w_{U,n}^{(i,j)}(z) = w_0^{(i,j)} \frac{r_0^{(i)}}{1 - r_0^{(i)}} \psi_{n,z}(x_0^{(i,j)}), \quad (9)$$

$$\tilde{w}_{U,n}^{(i,j)}(z) = \frac{w_{U,n}^{(i,j)}(z)}{\sum_{i=1}^{M_0} \sum_{j=1}^{L_0^{(i)}} \tilde{w}_{U,n}^{(i,j)}(z)}, \quad (10)$$

$$r_{L,n}^{(i)} = r_n^{(i)} \frac{1 - q_{L,n}^{(i)}}{1 - r_0^{(i)} q_{L,n}^{(i)}}, \quad (11)$$

$$p_{L,n}^{(i)}(x) = \sum_{j=1}^{L_0^{(i)}} \tilde{w}_{L,n}^{(i,j)} \delta_{x_0^{(i,j)}}(x), \quad (12)$$

$$r_{U,n}(z) = \frac{\sum_{i=1}^{M_0} r_0^{(i)} (1 - r_0^{(i)}) q_{U,n}^{(i)}(z)}{\kappa_n(z) + \sum_{i=1}^{M_0} \frac{r_0^{(i)} q_{U,n}^{(i)}(z)}{1 - r_0^{(i)} q_{L,n}^{(i)}}}, \quad (13)$$

$$p_{U,n}(x; z) = \sum_{i=1}^{M_0} \sum_{j=1}^{L_0^{(i)}} \tilde{w}_{U,n}^{(i,j)}(z) \delta_{x_0^{(i,j)}}(x). \quad (14)$$

**Table 1.** Definitions of metrics used in initialization per node [4].

$p_{D,n}$	detection probability
$g_n(\cdot   x)$	likelihood function
$\psi_{n,z}$	$g_n(\cdot   x) p_{D,n}(x)$
$\kappa_n(\cdot)$	Poisson clutter intensity

### 2.2. Downward Aggregation

Consider a parent node P and one of its child (leaf) node C. Node P passes multi-target state likelihood  $g_P(Z_P | \cdot)$  to node C that has maintained  $X_C^{(i,j)} \sim \{r_C^{(i)}, p_C^{(i)}(\cdot)\}_{i=1}^{M_C}$ . Node C has particles and weights  $\{w_C^{(i,j)}, x_C^{(i,j)}\}_{j=1}^{L_C^{(i)}}$ , which are used to compute weights  $\tilde{w}_C^{(i,j)}$ .

$$\tilde{w}_C^{(i,j)} = g_P(Z_P | X_C^{(i,j)}) g_C(Z_C | X_C^{(i,j)}) w_C^{(i,j)}, \quad (15)$$

$$\tilde{w}_C^{(i,j)} = \frac{\tilde{w}_C^{(i,j)}}{\sum_{j=1}^{L_C^{(i)}} \tilde{w}_C^{(i,j)}}. \quad (16)$$

Then node C proceeds to resample  $\{\tilde{w}_C^{(i,j)}, x_C^{(i,j)}\}_{j=1}^{L_C^{(i)}}$  by Epanechnikov kernel (with appropriately combined kernel functions and well-chosen kernel bandwidth)  $K_e(\cdot | \{\tilde{w}_C^{(i,j)}, x_C^{(i,j)}\}_{j=1}^{L_C^{(i)}})$  [2] and updates track probability, particles and weights.

$$x_{C,new}^{(i,j)} \sim K_e\left(\cdot \left| \{\tilde{w}_C^{(i,j)}, x_C^{(i,j)}\}_{j=1}^{L_C^{(i)}} \right.\right), j = 1, \dots, L_C^{(i)} \quad (17)$$

$$w_{C,new}^{(i,j)} = \frac{\tilde{w}_C^{(i,j)} p_{S,C}(x_C^{(i,j)})}{K_e(x_{C,new}^{(i,j)} | x_C^{(i,j)})} \quad (18)$$

$$\tilde{w}_{C,new}^{(i,j)} = \frac{w_{C,new}^{(i,j)}}{\sum_{j=1}^{L_C^{(i)}} w_{C,new}^{(i,j)}} \quad (19)$$

$$r_{C,new}^{(i)} = r_C^{(i)} \sum_{j=1}^{L_C^{(i)}} \tilde{w}_{C,new}^{(i,j)} p_{S,C}(x_C^{(i,j)}), \quad (20)$$

$$p_{C,new}^{(i)}(x) = \sum_{j=1}^{L_C^{(i)}} \tilde{w}_{C,new}^{(i,j)} \delta_{x_{C,new}^{(i,j)}}(x) \quad (21)$$

The weights and track probability  $r_{i,C}^{(i)}$  are then adapted as follows.

$$q_{I,C}^{(i)} = \sum_{j=1}^{L_C^{(i)}} w_{C,new}^{(i,j)} p_{D,C}(x_{C,new}^{(i,j)}), \quad (22)$$

$$w_{I,C}^{(i,j)} = w_{C,new}^{(i,j)} (1 - p_{D,C}(x_{C,new}^{(i,j)})), \quad (23)$$

$$\tilde{w}_{I,C}^{(i,j)} = \frac{w_{I,C}^{(i,j)}}{\sum_{j=1}^{L_C^{(i)}} w_{I,C}^{(i,j)}}, \quad (24)$$

$$r_{I,C}^{(i)} = r_{C,new}^{(i)} \frac{1 - q_{I,C}^{(i)}}{1 - r_{C,new}^{(i)} q_{I,C}^{(i)}}, \quad (25)$$

$$p_{I,C}^{(i)}(x) = \sum_{j=1}^{L_C^{(i)}} \tilde{w}_{I,C}^{(i,j)} \delta_{x_{C,new}^{(i,j)}}(x). \quad (26)$$

To make the mean cardinality balanced through the aggregation process, we take a heuristic approach to compare  $\sum_{i=1}^{M_C} r_C^{(i)}$  and  $\sum_{i=1}^{M_C} r_{I,C}^{(i)}$ . Rescaling of  $r_{I,C}^{(i)}$  may be taken when it is necessary to equalize  $\sum_{i=1}^{M_C} r_C^{(i)}$  and  $\sum_{i=1}^{M_C} r_{I,C}^{(i)}$ .

We summarize two metrics used in downward aggregation and defined in Table 2.

**Table 2.** Definitions of metrics used in downward aggregation.

$K_e(\cdot   \{\tilde{w}_C^{(i,j)}, x_C^{(i,j)}\}_{j=1}^{L_C^{(i)}})$	Epanechnikov kernel density
$p_{S,C}$	target existence probability

### 2.3. Upward Aggregation

As downward aggregation reaches the tree bottom, the terminal leaf node may pass messages backward. Consider a parent node P and the set of its child (lead) nodes  $\mathcal{C}$ . Node P has maintained  $X_P^{(i,j)} \sim \{r_P^{(i)}, p_P^{(i)}(\cdot)\}_{i=1}^{M_P}$  with particles and weights particles and weights  $\{w_P^{(i,j)}, x_P^{(i,j)}\}_{j=1}^{L_P^{(i)}}$ , and aggregates multi-target state likelihood  $g_q(Z_p | X_q^{(i,j)}), q \in \mathcal{C}$ . The joint likelihood  $\prod_{q \in \mathcal{C}} g_q(Z_q | X_P^{(i,j)})$  are used to compute weights  $\hat{w}_P^{(i,j)}$ .

$$\hat{w}_P^{(i,j)} = g_P(Z_P | X_P^{(i,j)}) \prod_{q \in \mathcal{C}} g_q(Z_q | X_P^{(i,j)}) w_P^{(i,j)}, \quad (27)$$

$$\tilde{w}_P^{(i,j)} = \frac{\hat{w}_P^{(i,j)}}{\sum_{j=1}^{L_P^{(i)}} \hat{w}_P^{(i,j)}}. \quad (28)$$

Node P then re-samples  $\{\tilde{w}_P^{(i,j)}, x_P^{(i,j)}\}_{j=1}^{L_P^{(i)}}$  by properly defined Epanechnikov kernel  $K_e(\cdot | \{\tilde{w}_P^{(i,j)}, x_P^{(i,j)}\}_{j=1}^{L_P^{(i)}})$  (with appropriately synergized kernel functions and well-chosen kernel bandwidth) and updates track probability, particles and weights.

$$x_{P,new}^{(i,j)} \sim K_e\left(\cdot | \{\tilde{w}_P^{(i,j)}, x_P^{(i,j)}\}_{j=1}^{L_P^{(i)}}\right), j = 1, \dots, L_P^{(i)} \quad (29)$$

$$w_{P,new}^{(i,j)} = \frac{\tilde{w}_P^{(i,j)} p_{S,P}(x_P^{(i,j)})}{K_e(x_{P,new}^{(i,j)} | x_P^{(i,j)})} \quad (30)$$

$$\tilde{w}_{P,new}^{(i,j)} = \frac{w_{P,new}^{(i,j)}}{\sum_{j=1}^{L_P^{(i)}} w_{P,new}^{(i,j)}} \quad (31)$$

$$r_{P,new}^{(i)} = r_P^{(i)} \sum_{j=1}^{L_P^{(i)}} \tilde{w}_{P,new}^{(i,j)} p_{S,P}(x_P^{(i,j)}). \quad (32)$$

$$p_{P,new}^{(i)}(x) = \sum_{j=1}^{L_P^{(i)}} \tilde{w}_{P,new}^{(i,j)} \delta_{x_{P,new}^{(i,j)}}(x) \quad (33)$$

The weights and track probability  $r_P^{(i)}$  are then adapted as follows.

$$q_{I,P}^{(i)} = \sum_{j=1}^{L_P^{(i)}} w_{P,new}^{(i,j)} p_{D,P}(x_{P,new}^{(i,j)}), \quad (34)$$

$$w_{I,P}^{(i,j)} = w_{P,new}^{(i,j)} (1 - p_{D,P}(x_{P,new}^{(i,j)})), \quad (35)$$

$$\tilde{w}_{I,P}^{(i,j)} = \frac{w_{I,P}^{(i,j)}}{\sum_{j=1}^{L_P^{(i)}} w_{I,P}^{(i,j)}}, \quad (36)$$

$$r_{I,P}^{(i)} = r_P^{(i)} \frac{1 - q_{I,P}^{(i)}}{1 - r_P^{(i)} q_{I,P}^{(i)}}, \quad (37)$$

$$p_{I,P}^{(i)}(x) = \sum_{j=1}^{L_P^{(i)}} \tilde{w}_{I,P}^{(i,j)} \delta_{x_{P,new}^{(i,j)}}(x). \quad (38)$$

Rescaling of  $r_{I,P}^{(i)}$  may be taken when it is necessary to balance  $\sum_{i=1}^{M_P} r_P^{(i)}$  and  $\sum_{i=1}^{M_P} r_{I,P}^{(i)}$ .

### 3. DYNAMIC ASPECTS IN INITIALIZATION

The framework for initialization in previous sections may be generalized to take into account time-evolving aspects as in typical RFS Bayesian filtering. We bring about the prerequisites of describing multi-target state dynamics and enriching the initialization process. The ingredients of formulating Bayesian recursion are  $X_k$  and  $Z_k$  that evolve over time instance  $k$ . Without pruning to eliminate insignificant sets, we always have  $X_k$  and  $Z_k$  absorb, in the sense of set union, new components with time. Let  $S_{k|k-1}(x_{k-1})$  denote the transition behavior of multi-target states that survive from time  $k$  to time  $k-1$ ,  $\Gamma_k$  denote ‘‘new-birth’’ states at time  $k$ , and  $U_k$  denote ‘‘measurement-corrected’’ states at time  $k$ . Thus,  $X_k$  represents the set union  $X_k = [\cup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1})] \cup \Gamma_k \cup U_k$  [2, 4].

Suppose  $X_{k-1} \sim \{r_{k-1}^{(i)}, p_{k-1}^{(i)}(\cdot)\}_{i=1}^{M_{k-1}}$  and  $X_k \sim \{r_k^{(i)}, p_k^{(i)}(\cdot)\}_{i=1}^{M_k}$ . Then one can conclude  $X_{k|k-1} \sim \{r_{S,k|k-1}^{(i)}, p_{S,k|k-1}^{(i)}(\cdot)\}_{i=1}^{M_{k-1}} \cup \{r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)}(\cdot)\}_{i=1}^{M_{\Gamma,k}}$ , and  $X_k \sim \{r_{S,k|k-1}^{(i)}, p_{S,k|k-1}^{(i)}(\cdot)\}_{i=1}^{M_{k-1}} \cup \{r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)}(\cdot)\}_{i=1}^{M_{\Gamma,k}} \cup \{r_{U,k}(z), p_{U,k}(z)\}_{z \in Z_k}$ . The cardinality evolves as  $M_k = M_{k-1} + M_{\Gamma,k} + |Z_k|$  [2, 4]. Let  $\Theta_k(x_k)$  denote the ‘‘normally detected’’ observations, and  $K_k$  denote clutter observation set at time  $k$ . The measurement set at time  $k$ ,  $Z_k$ , represents the following set union  $Z_k = [\cup_{x_k \in X_k} \Theta_k(x_k)] \cup K_k$ . At the beginning of recursive RFS Bayesian initialization, the filter takes as input the multi-target posterior density  $\pi_{k-1}(\cdot | Z_{1:k-1})$  at time  $k$ , and proceeds according to the prediction step in (39) and the update step in (40). We note that here  $k$  takes negative values to denote the time instances of pre-runs in initialization [2].

$$\pi_{k|k-1}(X_k | Z_{1:k-1}) = \int f_{k|k-1}(X_k | X) \pi_{k-1}(X | Z_{1:k-1}) \delta X \quad (39)$$

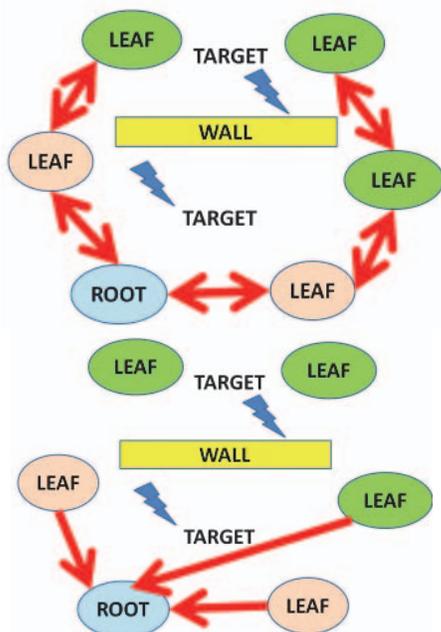
$$\pi_k(X_k | Z_{1:k}) = \frac{g_k(Z_k | X_k) \pi_{k|k-1}(X_k | Z_{1:k-1})}{\int g_k(Z_k | X) \pi_{k|k-1}(X | Z_{1:k-1}) \delta X} \quad (40)$$

### 4. SIMULATION RESULTS

We have examined the efficacy of our method to initialize the RFS posterior function of multi-target states over a sensor tree. (More details may be found in [5]). Here only target position states are focused, and the effects on velocity states are omitted. Each node initially maintains 100 particles for each track, which has track probability 0.9. Fig. 2 shows two scenarios: a sensor tree and a

one-hop only structure. One target is located in front of the wall (or below the “WALL” in Fig. 2). The other target is located behind the wall (or above the “WALL” in Fig. 2). The one-hop only structure fails to find the hidden target behind the wall (Fig. 2, right), but enables the root to initialize states of one target close the root and clutters (Fig. 3, right). The appearance of clutters is due to insufficient number of sensors participating in aggregation. Although the network discovers “probable targets” behind the wall, its inference introduces ambiguity owing to the leaf nodes not being able to provide reliable posterior functions.

A sensor tree supports multi-hop communication (Fig. 2, left) and may resolve two target states during initialization (Fig. 3, left). We note that in Fig. 2, many leaf nodes are not shown. Each node may have multiple leaf nodes, which sustains the initialization process even when some leaf nodes fail to pass messages. We consider appropriate thresholding, pruning, re-scaling and smoothing (e.g. with Epanechnikov kernel) over particles and weights to suppress minor peaks.

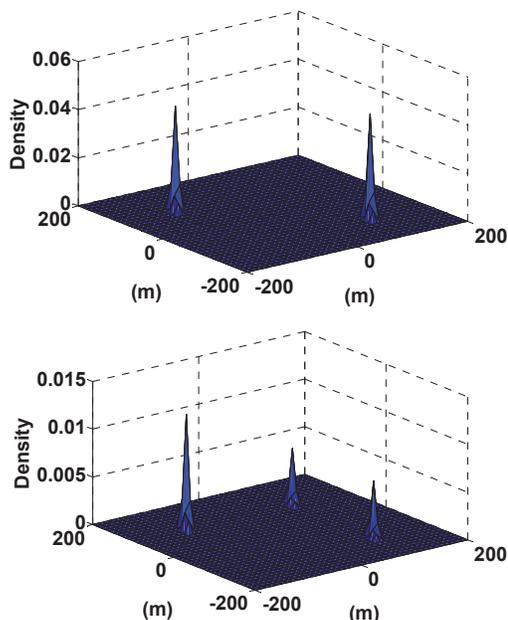


**Fig. 2.** A tree structure (top) supports multi-hop sensor aggregation. A one-hop structure (bottom) only supports the root and its immediate neighbors. Red arrows indicate established communication links. In a tree structure, a link supports downward and upward directions, when necessary, but not simultaneously. In the one-hop structure, a link only supports upward direction for aggregation.

## 5. CONCLUSIONS

We have developed a framework for initializing RFS type posterior functions over a sensor tree, which presents more challenges that are due to the need to update track probabilities, particles and weights in a more comprehensive manner. We design downward and upward aggregation steps that enable the root to launch initialization and have inference messages traverse from root to tree bottom and vice versa. Following the initialization process, each node refines its multi-target RFS density through one-hop communication with its neighbors (parent node and/or leaf nodes).

In a tree structure, a link supports downward and upward directions, when necessary, but not simultaneously. In the one-hop structure, a link only supports upward direction for aggregation. The tree structure allows the network to unveil targets hidden from obstacles, hence leveraging the reliability of the initialization process. In the one-hop structure, the leaf node only participates in the upward aggregation and acquires no knowledge of target states from ancestor nodes. More nodes, if not all, in a tree structure may gain knowledge from other nodes in the network and form a more comprehensive synergy among nodes to initialize particles and weights.



**Fig. 3.** A tree structure supports the root to initialize states of two targets (top). A one-hop structure may only allow the root to initialize states of one target and clutters (bottom).

## 6. REFERENCES

- [1] M. Borkar, V. Cevher, and J.H. McClellan, “Low computation and low latency algorithms for distributed sensor network initialization,” *Signal, Image and Video Processing Journal*, Springer, vol. 1, no. 2, , pp. 133-148, June 2007.
- [2] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, 2007.
- [3] W. -K. Ma, B. -N. Vo, S. Singh, and A. Baddeley, “Tracking an Unknown Time-Varying Number of Speakers using TDOA Measurements: A Random Finite Set Approach,” *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3291-3304, Sept., 2006.
- [4] B. -T. Vo, B. -N. Vo, and A. Cantoni, “The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409-423, 2009.
- [5] <http://tinyurl.com/6v92fnu>