RANDOM INFINITE TREE AND DEPENDENT POISSON DIFFUSION PROCESS FOR NONPARAMETRIC BAYESIAN MODELING IN MULTIPLE OBJECT TRACKING

Bahman Moraffah and Antonia Papandreou-Suppappola[†]

School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe Arizona 85281

Emails: bahman.moraffah@asu.edu, papandreou@asu.edu

ABSTRACT

Recent methods for tracking multiple objects have addressed important issues such as time-varying cardinality, unordered sets of measurements, and object labeling. Another challenge is how to robustly associate objects on a new scene with previously estimated objects. We propose a new method to track a dynamically varying number of objects using information from previously tracked ones. Our method is based on nonparametric Bayesian modeling using diffusion processes and random trees. We use simulations to demonstrate the performance of the proposed algorithm and compare it to a labeled multi-Bernoulli filter based tracker.

***Index Terms*—** Multiple object tracking, random tree, Dirichlet diffusion tree, nonparametric Bayesian modeling, sequential Monte Carlo method

1. INTRODUCTION

The multiple object tracking problem could include estimating the objects' time-varying cardinality, label and state parameters, among other information, depending on the application [1–8]. Among various approaches, random finite set methods were used to solve this problem, together with probability hypothesis density filtering and multi-Bernoulli or labeled multi-Bernoulli filtering [1, 3–5]. Nonparametric Bayesian methods were recently used for modeling evolving object state priors. In [9], the hierarchical Dirichlet process was used as a prior on the number of unobserved input modes to track manoeuvring objects. We recently used the dependent Dirichlet process to model the object prior and adaptively estimate both the object label and cardinality at each time step [10].

The Dirichlet diffusion trees (DDT), and its Pitman-Yor diffusion tree generalization, nonparametric Bayesian priors over tree structures, are thus useful for estimating latent parameters with hierarchical structure [11–13]. They were used, for example, in [14], as structure priors to infer different possible scenarios based on trees of different depth and path lengths. It was demonstrated in [15] that the high computational cost of Markov chain Monte Carlo (MCMC) inference can be avoided using efficient approximate inference DDT models. In this paper, we propose a dependent Poisson diffusion tree (D-PoDT) that extends the capability of DDTs to model hierarchies to also capture dependencies among the object states for the multiple object tracking problem. The dependent Poisson diffusion process (D-PoDP) introduces a prior on the space of the object state parameters using an infinite random tree. It is used as a state prior to capture the time-dependency among the states and estimate the state parameters. A time-dependent process is introduced to infer from the measurements and update the object state parameters, label the objects and and estimate the object cardinality at each time step. An MCMC sampling method integrates the distribution over the infinite random tree and the time-dependent process for updating the states.

The rest of the paper is organized as follows. Section 2 describes the time-varying multiple object tracking model. Section 3 provides detailed information on our proposed multiple object tracking algorithm. In Section 4, the performance of the algorithm is demonstrated using a simulated example with five objects with time-dependent trajectories.

2. TRACKING MODEL

We consider a multiple object tracking model with time-varying numbers of objects entering, leaving or remaining in the scene at each time step k. The object cardinality N_k and the number of measurements L_k are both assumed unknown [5, 10]. This tracking model is used to jointly estimate the object state information and the cardinality at each time step. We assume that the sample spaces of the ℓth object state vector $\mathbf{x}_{\ell,k}$, $\ell = 1, \ldots, N_k$ and lth measurement vector $\mathbf{z}_{l,k}$, $l = 1, \ldots, L_k$, at time step k, are $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, respectively. We also assume that the sequence $X_k = \{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,N_k}\}$ corresponds to the configuration of the multiple object state vectors at time step k.

Given the state vector configuration at time $(k-1)$, we consider three possible scenarios for the ℓth object and its state vector at time step k: (a) the object leaves the scene with probability $(1 - \mathrm{P}_{\ell,k|k-1})$; (b) the object remains in the scene with probability $\mathrm{P}_{\ell,k|k-1}$ and its state $\mathbf{x}_{\ell,k-1}$ transitions

[†]This work was supported in part by Grant AFOSR FA9550-17-1-0100.

with probability $\mathrm{P}_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{k-1})$ and unknown parameter vector $\boldsymbol{\theta}$; and (c) a new object, with state $\mathbf{x}_{\ell,k} \in X_k$, enters the scene generating a measurement. We also assume that each measurement is generated by only one object and that measurements are independent of one another.

3. DEPENDENT DIFFUSION MODELING

We propose a new method for multiple object tracking based on a D-PoDP. These are similar to the Dirichlet diffusion trees in [11] and Pitman-Yor diffusion trees in [12] in that they can be used as priors to latent parameters to capture hierarchical structure. The D-PoDTs are different, however, in that the prior can directly incorporate time-dependent learned information. For multiple object tracking, the state prior can include the number of objects at the current time and the object label at the previous time. Thus, the proposed method can be used to make inference on the object labels over related information by tracing random tree paths. Following details the proposed D-PoDP algorithm.

3.1. Poisson Diffusion Process

We consider a class of priors on trees whose terminal nodes (leaves) are the object state parameters, and whose non-terminal nodes (branch nodes) represent clustering of the state parameters in a hierarchy. We assume that a tree may have infinite number of vertices, and every edge can occur with some probability. The probability of an edge occurring that violates the tree conditions is assumed zero. We assume that the first vertex (at time step $k=0$) is drawn from $P_{\boldsymbol{\theta}_0}$ with probability 1. To generate this infinite random tree, the branch nodes and leaves must be specified. We describe the generative process in terms of a diffusion process on a unit interval; that is, the leaves correspond to the location of the diffusion process at time step $k=1$. Each point starts at time $k=0$ and follows a diffusion process, i.e., a Brownian motion, until time $k=1$, where it is observed. For example, we assume that the first object state at time step $k=1$, $\boldsymbol{\theta}_{1,1}$, is drawn from a diffusion process and fixed. The second object state, $\boldsymbol{\theta}_{1,2}$, starts at time $k=0$ and follows the same path as $\boldsymbol{\theta}_{1,1}$ up to time δt (time between steps) before it diverges from the first path and takes an independent path. The generative process for the ith object parameter at time step $k=1$ is as follows. At a branch point, if $\boldsymbol{\theta}_{1,i}$ does not diverge off the branch before reaching to the previous divergence point, then the previous branches are selected with probability

$$\mathrm{P}(\text{select } j\text{th path}) = \frac{n_j - \beta}{m+\eta}, \quad \mathrm{P}(\text{diverge}) = \frac{\eta - \beta K}{m+\eta}\,. \quad (1)$$

Here, n_j is the number of objects previously in the jth branch, K is the total number of branches originating from this branch point, $m = \sum_{l=1}^{K} n_l$, and β and η are discount and concentration hyperparameters. It was shown in [12] that, since the specific diffusion path taken between nodes can be ignored, the probability of generating a specific tree structure with associated divergence times can be determined by the accumulative divergence function $H(\cdot)$; this can analytically determine the locations at each leaf node. Therefore, $\boldsymbol{\theta}_{1,i}$ follows the path of the previous points and diverges in the interval δt, assuming it has not diverged up to time $t \in [0,1]$, with probability

$$\frac{\Gamma(m-\beta)}{\Gamma(m+1-\eta)} \int_{\delta t} dH(s)\,.$$

Here, Γ is the gamma function, m is the number of points that have previously traversed this path, and β and η are discount and concentration hyperparameters, respectively. For large m, the probability of diverging from this path is small. As a result, an infinite random tree can be generated from which there is a probability measure on each vertex that is dependent on its parent vertex. Note that the random tree generated is exchangeable [12]. Therefore, the probability of generating a specific tree, divergence times, and divergence locations is invariant to the ordering of the object state parameters.

The proposed algorithm is initialized by drawing N_1 from a Poisson distribution, $N_1 \sim \mathrm{Po}(\alpha)$ for some α. Subsequently, we select N_1 state parameters (leaves), which, without loss of generality, can be assumed to be the first N_1 leaves due to exchangeability. We then set $\{\boldsymbol{\theta}_{1,1}, \ldots, \boldsymbol{\theta}_{1,N_1}\}$ to be the first N_1 parameters generated through this process that are associated with the state configuration $\{\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{1,N_1}\}$.

3.2. Transition from Time $k-1$ to Time k

We define V_{k-1} and $V_{B,k-1}$ to be the set of generated state parameters (leaves) nodes and branch nodes, respectively, that are connected to the state parameter (leaf) node at time $k-1$. Each point $\boldsymbol{\theta}_{\ell,k-1} \in V_{k-1}$ that is generated in the tree has two options: (i) it can remain in the tree with probability $\mathrm{P}_{\ell,k|k-1}$ and transition to $\boldsymbol{\theta}_{\ell,k}$ according to the transition kernel probability $\nu(\boldsymbol{\theta}_{\ell,k-1}, \boldsymbol{\theta}_{\ell,k})$, which is proportional to the transition probability $\mathrm{P}_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{k-1})$; (ii) it can leave the tree with probability $(1-\mathrm{P}_{\ell,k|k-1})$. We assume that the following parameters are available at time $k-1$:

- N_{k-1}, number of objects
- $V_{k-1} = \{\boldsymbol{\theta}_{k-1,1}, \ldots, \boldsymbol{\theta}_{k-1,N_{k-1}}\}$, generated parameters
- $V_{B,k-1}$, branch nodes connected to a leaf node
- $V_{k|k-1} \subseteq V_{k-1}$, survived parameters
- $V_{B,k|k-1} \subseteq V_{B,k-1}$, survived branch nodes
- $S_{a,k-1}$, siblings with common parent branch node a
- $S_{a,k|k-1} \subseteq S_{a,k-1}$, survived siblings with common parent branch node a

Note that if all the leaves connected to a branch node disappear, the branch node is removed from the set of branch nodes. A probability vector $\mathbf{p}_{\text{branch}} = [p_a]_{a \in V_{B,k|k-1} \cup \delta}$ is then assigned to the survived branch nodes as

$$p_a = \begin{cases} \frac{|S_{a,k-1}|+|S_{a,k|k-1}|-\gamma}{N_{B,k|k-1}-1+\sum_{a\in V_{B,k|k-1}}|V_{a,k-1}|+\zeta}, & a \in V_{B,k|k-1} \\ \frac{\zeta-|V_{B,k|k-1}|\gamma}{N_{B,k|k-1}-1+\sum_{a\in V_{B,k|k-1}}|V_{a,k-1}|+\zeta}, & a = \delta \end{cases}$$

where $|S_{a,k-1}|$ is the cardinality of the set $S_{a,k-1}$, $N_{B,k|k-1}$ is the number of points that survives after transition, δ denotes a new branch, P_δ is the probability of generating a new branch, and ζ and γ are hyperparameters.

3.3. Evolution and Parameter Estimation at Time k

At time k, we utilize the distribution on set $V_{k|k-1}$ to find V_k. To this end, we can assume that $\boldsymbol{\theta}_{i,k|k-1} \in S_{a,k|k-1}$, $i=1,\dots,|V_{k|k-1}|$ are transitioned from time $k-1$ to k. We draw $\tilde{N}_{i,k|k-1}\sim\mathrm{Po}(\frac{p_a\times\alpha}{2\,|S_{a,k|k-1}|})$ and draw $\tilde{N}_{i,k|k-1}$ points given $\boldsymbol{\theta}_{i,k|k-1}$ based on a diffusion process described in Section 3.1. At time k, we also draw $\tilde{N}_{\delta,k|k-1} \sim \mathrm{Po}(\frac{p_\delta\times\alpha}{2})$ and draw $\tilde{N}_{\delta,k|k-1}$ new points from the infinite random graph from $P_{\boldsymbol{\theta}_0}$. We set $\tilde{N}_k = \Sigma_i \tilde{N}_{i,k|k-1}$ and $\tilde{V}_k = \{\theta_1,\dots\theta_{\tilde{N}_k}\}$. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 D-PoDP-EMM algorithm

```
Initialization:
• Draw $\boldsymbol{\theta}_0^0 \sim P_{\boldsymbol{\theta}_0}$
• Draw $N_1 \sim \mathrm{Po}(\alpha)$
• Generate $\{\boldsymbol{\theta}_{1,1},\dots,\boldsymbol{\theta}_{1,N_1}\}$ based on a diffusion process
  with branching probability of convergence in (1)
Transitioning from time $k-1$ to $k$
for $\boldsymbol{\theta}_{i,k|k-1} \in V_{k|k-1}$ do
    Draw $\tilde{N}_{i,k|k-1} \sim \mathrm{Po}(\frac{p_a\times\alpha}{2\,|S_{a,k|k-1}|})$
    Generate $\tilde{N}_{i,k|k-1}$ parameter points given $\boldsymbol{\theta}_{i,k|k-1}$ using
      a diffusion process
end for
• Draw $\tilde{N}_{\delta,k|k-1} \sim \mathrm{Po}(\frac{p_\delta\times\alpha}{2})$
• Draw $\tilde{N}_{\delta,k|k-1}$ new parameter points from the base distribution
  $P_{\boldsymbol{\theta}_0}$ following a diffusion process
At time $k$
Set $\tilde{N}_k = \sum_i \tilde{N}_{i,k|k-1}$
Set $\tilde{V}_k = \{\boldsymbol{\theta}_{k,1},\dots,\boldsymbol{\theta}_{k,\tilde{N}_k}\}$.
Set $X_k = \{\mathbf{x}_{k,1},\dots\mathbf{x}_{k,\tilde{N}_k}\}$.
```

3.4. Inference through a Dependent Mixture Model

The D-PoDP in Algorithm 1 provides a joint estimation of the object state parameters and number of objects, at time step k. At time k, the measurement vector, $\mathbf{z}_{l,k}$, $l=1,\dots,L_k$, becomes available to update the time-dependent cardinality and infer the posterior distribution. Note that the probability of selecting some of the generated parameters may be zero; also, some new parameters may also need to be generated. We introduce an algorithm to dependently cluster these measurements as follows.

We use the state parameter vector distribution from the output of Algorithm 1 as the mixing distribution to infer measurement distributions to update the object cardinality. The probability of choosing a parameter $\boldsymbol{\theta}_{i,k}$ is proportional to the popularity of the parameter at time k, in addition to the cardinality of the set of siblings with the common parent branch node at time $k-1$. Specifically, if we assume that $\boldsymbol{\theta}_{\ell,k}$ is transitioned from $\boldsymbol{\theta}_{\ell,k-1}$ (for which it shares the common parent a), then $\pi_\ell = \mathrm{P}(\text{select } \boldsymbol{\theta}_{\ell,k}) \propto (n_{\ell,k} + |S_{a,k-1}|)$, where $n_{\ell,k}$ is the number of measurements that have already selected $\boldsymbol{\theta}_{\ell,k}$ at time k. The probability of selecting a parameter that has not been used up to time k is proportional to some hyperparameter λ. In particular, $p(\mathbf{z}_{l,k} \mid \mathbf{x}_{\ell,k}, \boldsymbol{\theta}_{\ell,k}, \pi_\ell)$ can be inferred as

$$\pi_\ell \propto \begin{cases} n_{\ell,k} + |S_{a,k-1}|, & \theta_{\ell,k-1} \in S_{a,k-1}, \theta_{\ell,k} \in \tilde{V}_k \\ \lambda, & \textit{New} \;\; \theta_{\ell,k} \end{cases} \tag{2}$$

$$\mathbf{x}_{\ell,k} \mid \boldsymbol{\theta}_{\ell,k} \sim G(\theta_{\ell,k}) \tag{3}$$

$$\mathbf{z}_{\ell,k} \mid \mathbf{x}_{\ell,k}, \boldsymbol{\theta}_{\ell,k}, \pi_\ell \sim F(\mathbf{x}_{\ell,k}, \boldsymbol{\theta}_{\ell,k}) \tag{4}$$

where G and F are two appropriately selected distributions. Algorithm 2 summarizes the implementation of the dependent mixtures to cluster the measurements and track the objects. Note that since the D-PoDP is used to find the object trajectories, one needs to trace the random tree. Algorithms 1 and 2, together with MCMC sampling methods, constitute the proposed D-PoDP multiple object tracking algorithm. Sampling in both algorithms is performed using MCMC methods; in particular, we use Gibbs sampling for models based on conjugate prior distributions.

Algorithm 2 Dependent mixture model to cluster measurements and track objects

```
Input: Measurements: $\{\mathbf{z}_{1,k},\dots,\mathbf{z}_{k,L_k}\}$
Output: $N_k$, cluster configurations, and posterior
At time k
Sample $\{\boldsymbol{\theta}_{1,k},\dots,\boldsymbol{\theta}_{k,\tilde{N}_k}\}$ and $\{\mathbf{x}_{k,1},\dots,\mathbf{x}_{k,\tilde{N}_k}\}$
according to Algorithm 1
Draw $\{\pi_i\}$ according to (2)
for $l = 1$ to $L_k$ do
    Sample $\mathbf{z}_{l,k}|\mathbf{x}_{\ell,k},\boldsymbol{\theta}_{\ell,k},\pi_\ell$ using (4)
end for
$N_k \leftarrow \tilde{N}_k$
$V_k = \{\boldsymbol{\theta}_{1,k},\dots,\boldsymbol{\theta}_{N_k,k}\}$
return $N_k$ and posterior of $\mathbf{z}_{l,k}|\mathbf{x}_{\ell,k},\boldsymbol{\theta}_{\ell,k},\pi_\ell$
```

4. SIMULATION RESULTS

In order to demonstrate the performance of our proposed D-PoPD method, we simulated a dynamic nonlinear tracking ex-

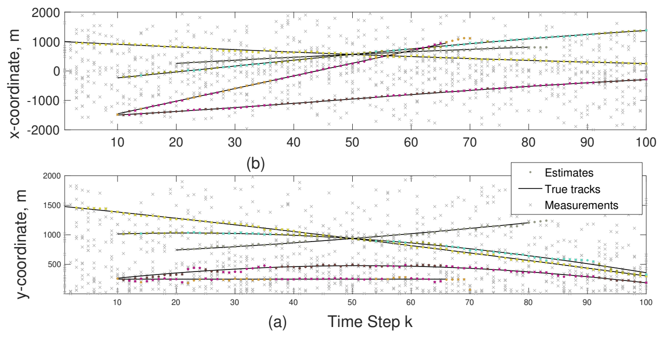

Fig. 1: True and estimated (a) x- and (b) y-coordinates as a function of the time step k of five objects.

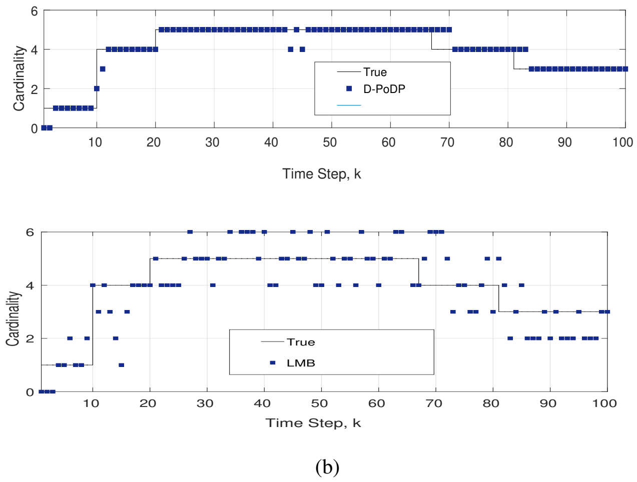

Fig. 2: Comparison of estimated cardinality using (a) proposed D-PoDP method and (b) LMB when tracking 5 objects.

ample using five objects that enter and leave a scene at different times. The overall observed time is $K = 100$ times steps and the signal-to-noise ratio (SNR) was -3 dB. Th time steps over which each object is present in the scene are summarized in Table 1. The time steps are also depicted in Figures 1(a) and (b) that show that x and y-coordinates of the true trajectory of each object. The D-PoPD estimated x and y-coordinates of the trajectory of each object are also shown in Figures 1(a) and (b). The D-PoPD algorithm was com-

Table 1: Time step at which object enters and leaves the scene

Object	Time Enters	Time Leaves
Object 1	$k = 0$	$k = 100$
Object 2	$k = 10$	$k = 100$
Object 3	$k = 10$	$k = 100$
Object 4	$k = 10$	$k = 60$
Object 5	$k = 20$	$k = 80$

pared with the labeled multi-Bernoulli (LMB) based tracker; both algorithms used 10,000 Monte Carlo (MC) simulations. As shown in Figures 2a and 2b, the proposed tracker is more accurate in estimating the time-dependent object cardinality than the LMB. This is also demonstrated using the optimal sub-pattern assignment (OSPA) metric (of order $p = 1$ and cut-off $c = 100$) for range an cardinality in Figures 3(a) and 3(b), respectively. As it can be seen for the D-PoPD, for example, for the cardinality OSPA measure, the highest error is observed at time step $k = 0$, when the first object enters the scene and then at time step $k = 10$, when three new objects enter the scene. The method performs very well for a long time, tracking all four objects in the scene, with only a small error that soon decreases object 5 enters the scene. It continues to track the correct number of objects even when object 4 leaves the scene.

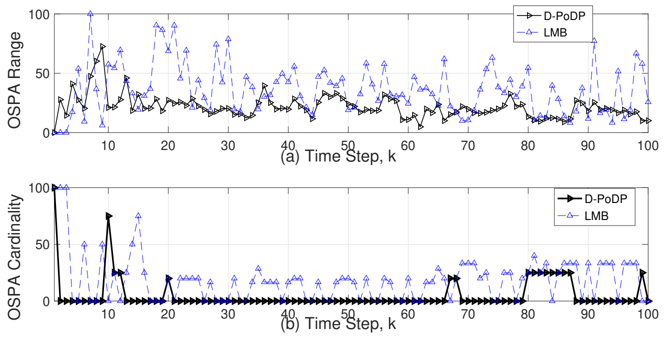

Fig. 3: OSPA of order $p = 1$ and cut-off $c = 100$ for (a) range and (b) cardinality averaged over 10,000 MC simulations for the proposed D-PoPD and the labeled multi-Bernoulli (LMB) based tracker.

5. CONCLUSION

In this paper, we presented a novel class of random trees to address the multiple object tracking problem by using a diffusion process and random trees where tracing the tree allows for tracking object trajectories. We demonstrated that the proposed dependent Poisson diffusion process with Bayesian nonparametrics modeling and multiple object tracking can efficiently obtain object tracks, labels and time-varying cardinality. Moreover, the sequential Monte Carlo implementation of the proposed tracking framework verifies the accuracy and simplicity of this algorithm.

6. REFERENCES

[1] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, Inc., 2007.

[2] W. Koch, *Tracking and Sensor Data Fusion*, Springer, 2016.

[3] 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, pp. 409–423, 2009.

[4] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 62, pp. 3246–3260, 2014.

[5] B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne III, R. Mahler, and B.-T Vo, "Multitarget tracking," *Wiely Encyclopedia of Electrical Eng.*, 2015.

[6] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Conference on Computer Vision and Pattern Recognition*, 1999, pp. 253–259.

[7] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull, "Object tracking by particle filtering techniques in video sequences," in *Advances and Challenges in Multisensor Data and Information*, E. Lefebvre, Ed., vol. 8, pp. 260–268. IOS Press, 2007.

[8] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "Autonomous robot navigation in highly populated pedestrian zones," *Journal of Field Robotics*, vol. 32, pp. 565–589, 2015.

[9] E. B. Fox, E. B. Sudderth, and A. S. Willsky, "Hierarchical Dirichlet processes for tracking maneuvering targets," in *International Conference on Information Fusion*, 2007, pp. 1–8.

[10] B. Moraffah and A. Papandreou-Suppopola, "Dependent Dirichlet process modeling and identity learning for multiple object tracking," in *Asilomar Conference on Signals, Systems, and Computers*, 2018.

[11] R. M. Neal, "Density modeling and clustering using Dirichlet diffusion trees," in *Bayesian Statistics 7*, J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, et al., Eds., pp. 619–629. Oxford University Press, 2003.

[12] D. A. Knowles and Z. Ghahramani, "Pitman-Yor diffusion trees," in *Conference in Uncertainty in Artificial Intelligence*, 2011, pp. 410–418.

[13] Z. Ghahramani, "Bayesian non-parametrics and the probabilistic approach to modelling," *Philosophical Transactions of the Royal Society*, p. 20 pgs, 2013.

[14] E. W. Meeds, D. A. Ross, R. S. Zemel, and S. T. Roweis, "Learning stick-figure models using nonparametric Bayesian priors over trees," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] D. A. Knowles, J. V. Gael, and Z. Ghahramani, "Message passing algorithms for Dirichlet diffusion trees," in *International Conference on Machine Learning*, 2011.