A Multi-Sensor, Gibbs Sampled, Implementation of the Multi-Bernoulli Poisson Filter

Leonardo Cament*, Martin Adams†, Javier Correa‡
Department of Electrical Engineering & Advanced Mining Technology Center, Universidad de Chile
Av. Tupper 2007, Santiago, Chile
Email: *lcament@ing.uchile.cl, †martin@ing.uchile.cl, ‡javier.correa@amtc.cl

***Abstract*—This paper introduces and addresses the implementation of the Multi-Bernoulli Poisson (MBP) filter in multi-target tracking. A performance evaluation in a real scenario, in which a 3D lidar, automotive radar and a video camera are used for tracking people will be provided. For implementation purposes, a Gaussian Mixture (GM) approximation of the MBP filter is used. Comparisons with state of the art GM-δ-GLMB and GM-δ-GMBP filters show similar accuracy, despite the need for less parameters, and therefore less computational cost, within the GM-MBP filter. Further performance improvements of the GM-MBP filter are shown, based on birth intensity and survival distributions, which take into account the common field of view of the sensors and the variation of time steps between asynchronous measurements.**

***Index Terms*—random finite sets, multi-target tracking, multi-Bernoulli filter, faster R-CNN**

I. Introduction

In the field of multi-target tracking, Random Finite Set (RFS) based algorithms have recently offered robust solutions [1].

In a manner analogous to which the state of the art Labeled Multi-Bernoulli (LMB) filter [2] is related to the δ-Generalized Labeled Multi-Bernoulli (δ-GLMB) filter [3], the Multi-Bernoulli Poisson (MBP) filter used in this article is a derivative of the δ-Generalized Multi-Bernoulli Poisson (δ-GMBP) filter presented in [4], [5][1].

The LMB filter [2] requires the intermediate conversion and reconversion of LMB to δ-GLMB components for filter updates. In contrast, due to Gibbs sampling, the MBP and Gibbs-LMB filters [6], compute the parameters of the posterior directly from the prior distribution, without the necessity of such conversions, significantly decreasing their computational complexity. Also, in a manner similar to the δ-GLMB filter, the δ-GMBP filter uses a multi-Bernoulli RFS for the detected targets. However, in contrast to the δ-GLMB filter, the δ-GMBP filter adopts a Poisson RFS for the birth process. In contrast to a multi-Bernoulli birth RFS, a Poisson birth RFS imposes no restriction on the number of birth targets. For these reasons, the MBP filter is implemented in this article.

The δ-GMBP RFS is a combination of a δ-Generalized Multi-Bernoulli (δ-GMB) RFS and a Poisson RFS, and is closed under prediction and update. The δ-GMB RFS models the known targets, while the Poisson RFS models the potential targets that have not yet been detected. The MBP filter is computationally cheaper than the δ-GMBP filter because it does not maintain the correlation information between the target to measurement associations.

This article focuses on a Gaussian Mixture (GM)-MBP filter implementation with asynchronous multi-sensor measurements. Experiments are carried out in an urban scenario in which people are tracked using a 3D lidar, a 2D radar and a video camera. The detections of people are corrupted by vehicles, trees and other artefacts.

For benchmark purposes, performance comparisons are made between the GM-MBP and the δ-GLMB and δ-GMBP filters, based on the same detection statistical models and environment. To further improve the performance of the GM-MBP filter, the variation of time steps between the asynchronous measurements is used to modify the single-target motion model and probability of survival and birth intensity.

Section II provides a brief introduction to RFS based multi-target tracking and the theory behind the MBP filter and Section III presents its GM implementation. Section IV describes the sensors, people detectors and motion models and comparative results are presented in Section V.

II. Theoretical Background

A. Random Finite Sets Overview

An RFS is a set containing a finite number of random variables which can also be an empty set. It is random in the number of elements and in the values of each element, and it is described by its Probability Density Function (PDF). Poisson and multi-Bernoulli are common RFS distribution types. The Poisson RFS models the number of elements in the set following a Poisson distribution, while the elements are spatially distributed according to a given density function. A multi-Bernoulli RFS models the existence or non-existence of the elements, and when the elements exist they distribute according to a given PDF.

B. Standard Bayesian Recursive Filtering

Bayesian filtering consists of two parts. The prediction follows the Chapman-Kolmogorov equation:

$$p(X_t|Z_{1:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})\delta X_{t-1}, \quad (1)$$

[1]Note that in [4] the δ-GMBP filter was referred to as the Dirac Delta Mixture with Poisson (DMP) filter, and renamed as δ-GMBP in [5].

978-0-9964527-7-9 ©2018 ISIF

while the update uses Bayes rule:

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t)p(X_t|Z_{1:t-1}), \tag{2}$$

where X_{t-1} and X_t represent the multi-target state at time steps $t-1$ and t respectively, $Z_{1:t-1}$ all observed measurements from time 1 to $t-1$, $p(X_t|X_{t-1})$ the state transition model and $p(Z_t|X_t)$ the measurement model.

In order to solve Equations (1) and (2) the transition and measurement models must fulfill various properties [7, p. 313].

C. The Multi-Bernoulli Poisson (MBP) Filter

1) Single Target MBP Prediction: Assume X_{t-1} is the prior state RFS and $f_x(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the standard single target transition model [7], where $\mathbf{x}_t$ is the single target state vector. Assume also that X_{t-1} is modeled with a multi-Bernoulli RFS with parameters $\bigcup_{n=1}^{N}\{(r_{t-1}^n, \theta_{t-1}^n)\}$, where r_{t-1}^n is the probability of existence and θ_{t-1}^n are the parameters representing the state distribution $f(\mathbf{x};\theta_{t-1}^n)$, and N is the total number of MBP components. Then the predicted distribution for X_t is also a multi-Bernoulli RFS with parameters obtained from:

$$r_{t|t-1}^n = r_{t-1}^n \int \langle f_x(\mathbf{x}|\cdot), P_s(\cdot)f(\cdot;\theta_{t-1}^n)\rangle d\mathbf{x} \tag{3}$$

$$f(\mathbf{x};\theta_{t|t-1}^n) = \frac{\langle f_x(\mathbf{x}|\cdot), P_s(\cdot)f_{t-1}(\cdot;\theta_{t-1}^n)\rangle}{\int\langle f_x(\mathbf{x}|\cdot), P_s(\cdot)f_{t-1}(\cdot;\theta_{t-1}^n)\rangle d\mathbf{x}} \tag{4}$$

where $\langle f, g\rangle = \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$, and $P_s(\cdot)$ represents the target survival probability distribution [4].

2) Joint MBP Prediction and Update: An efficient implementation of the δ-GLMB filter was proposed by Vo et al. [8] using the Gibbs sampler. In a similar way the Gibbs sampler is used in this work, but instead of using the weights of the posterior δ-GLMB components [6], a histogram built by counting each sampled data association is used, as described below.

In order to implement the Gibbs sampler, the cost function for the MBP is represented by the weights of the mixture model presented by Williams in [9]:

$$\eta_n(m) = \begin{cases} 1 - r_{t|t-1}^n + \\ \quad r_{t|t-1}^n\langle 1 - P_D(\cdot), f(\cdot;\theta_{t|t-1}^n)\rangle & \text{if } m = 0 \\ \dfrac{r_{t|t-1}^n\langle P_D(\cdot)f(\cdot;\theta_{t|t-1}^n), f_z(\mathbf{z}_m|\cdot)\rangle}{\kappa(\mathbf{z}_m) + \langle P_D(\cdot)D_{B,t}(\cdot), f_z(\mathbf{z}_m|\cdot)\rangle}, & \text{if } m \geq 1 \end{cases}. \tag{5}$$

where $P_D(\cdot)$ is the target detection probability distribution, $\kappa(\mathbf{z}_m)$ is the clutter intensity, and $D_{B,t}(\cdot)$ is the birth intensity.

The Gibbs sampler, with the cost function (5), is processed multiple times, and a histogram $h_{n,m}$ is built as a proportion of samples in which target[2] n was associated with measurement m. $\sum_{m=0}^{M} h_{n,m} = 1$ for $n > 0$ and $\sum_{n=0}^{N} h_{n,m} = 1$ for $m > 0$, where M is the total number of detections. $h_{n,0}$ represents the proportion of samples in which the target does not exist or is misdetected, and the proportion in which the measurement m was unassociated (equivalent to misdetection) is represented by $h_{0,m} = 1 - \sum_{n=1}^{N} h_{n,m}$.

The posterior existence probability and spatial distribution multi-Bernoulli parameters of the existing targets are given by:

$$\begin{aligned} r_{t|t}^n =& h_{n,0}\frac{r_{t|t-1}^n\langle 1 - P_D(\cdot), f(\cdot;\theta_{t|t-1}^n)\rangle}{1 - r_{t|t-1}^n + r_{t|t-1}^n\langle 1 - P_D(\cdot), f(\cdot;\theta_{t|t-1}^n)\rangle} \\ &+ \sum_{m=1}^{M} h_{n,m}, \end{aligned} \tag{6}$$

$$\begin{aligned} f(\mathbf{x};\theta_{t|t}^n) =& h_{n,0}f(\mathbf{x};\theta_{t|t-1}^n) + \\ & \sum_{m=1}^{M} h_{n,m}\frac{P_D(\mathbf{x})f(\mathbf{x};\theta_{t|t-1}^n)f_z(\mathbf{z}_m|\mathbf{x})}{\langle P_D(\cdot)f(\cdot;\theta_{t|t-1}^n), f_z(\mathbf{z}_m|\cdot)\rangle}. \end{aligned} \tag{7}$$

The posterior existence probability and spatial distribution of the new targets are given by:

$$r_{t|t}^{N+m} = h_{0,m}\frac{\langle P_D(\cdot)D_{B,t}(\cdot), f_z(\mathbf{z}_m|\cdot)\rangle}{\kappa(\mathbf{z}_m) + \langle P_D(\cdot)D_{B,t}(\cdot), f_z(\mathbf{z}_m|\cdot)\rangle}, \tag{8}$$

$$f(\mathbf{x};\theta_{t|t}^{N+m}) = \frac{P_D(x)D_{B,t}(\mathbf{x})f_z(\mathbf{z}_m|\mathbf{x})}{\langle P_D(\cdot)D_{B,t}(\cdot), f_z(\mathbf{z}_m|\cdot)\rangle}. \tag{9}$$

3) Adding labels to the state: The main difference between the LMB and the MBP RFSs is the birth distribution. Once the targets exist, both use the same Multi-Bernoulli distribution, as demonstrated in [10]. When using an LMB birth model, each LMB component is a possible new target. In contrast, when the birth is Poisson, each measurement produces a new possible target. In point target tracking, the target is associated with only one measurement. Thus, we label the new target (k, m), where m is the measurement index, and k the time step, as takes place in the δ-GLMB filter. After the update, the target becomes a Bernoulli component, maintaining the label in time as proved in [10]. It is important to emphasize that this manner of adding labels is possible for a single-point target because it is certain the target is produced by only one measurement, but not for an extended target because for example one measurement can produce two new targets (in different MB components) because the measurement belongs to different clusters of points.

III. Gaussian Mixture MBP Filter Implementation

The state distribution is represented by a GM, in which the parameters of the state are $\theta_t^n = \{(\omega_t^{n,1}, \boldsymbol{\mu}_t^{n,1}, \boldsymbol{\Sigma}_t^{n,1}), ..., (\omega_t^{n,N_t^n}, \boldsymbol{\mu}_t^{n,N_t^n}, \boldsymbol{\Sigma}_t^{n,N_t^n})\}$, where $\omega_t^{n,i}$ is the weight, $\boldsymbol{\mu}_t^{n,i}$ the mean vector, $\boldsymbol{\Sigma}_t^{n,i}$ the covariance matrix of the GM, $i \in \mathbb{N}, i \leq N_t^n$, where N_t^n represents the number of Gaussians in the GM. The state distribution for the nth component of the multi-Bernoulli RFS is given by:

$$f(\mathbf{x},\theta_t^n) = \sum_{i=1}^{N_t^n} \omega_t^{n,i}\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_t^{n,i}, \boldsymbol{\Sigma}_t^{n,i}). \tag{10}$$

Next, the birth of new hypotheses, state extraction and GM-MBP recursion approximations are described.

[2]The same variable n is used for target number and multi-Bernoulli component, since each component represents a target hypothesis.

1) Birth Hypotheses: The mixture weights of new targets are obtained from the Poisson component of the δ-GMBP RFS. Because a uniform spatial distribution is used, there is no closed form solution for computing this component, but a reasonable assumption is that the components can be obtained by sampling points from the inverse measurement likelihood function $\mathbf{x}_t \sim g_z^{-1}(\mathbf{z}_t, \cdot)$, where $\mathbf{z}_t$ is the measurement vector at time t. This concept is used in our experiments.

2) State Extraction: To extract the estimated state from the filter, the target hypothesis with a probability of existence greater than a determined threshold λ_1 is taken as a track. Hysteresis is used to maintain the track, when its probability of existence reduces. Thus, when an existing target reduces its probability of existence to a value lower than a second threshold λ_2 the track is considered nonexistent, i.e., $0 < \lambda_2 < \lambda_1 < 1$. The target position is then estimated by the weighted average GM state mean:

$$\bar{\mathbf{x}}_t^n = \sum_{i=1}^{N_n^t} \omega_t^{n,i} \boldsymbol{\mu}_t^{n,i}. \tag{11}$$

3) GM-MBP Recursion Approximations: The probabilities of detection $P_D(\cdot)$ and survival $P_s(\cdot)$ are assumed to be vary slowly, and are thus represented by a constant value in the vicinity of the states. For a GM state representation $P_D(\cdot) \approx P_D(\mu)$ and $P_s(\cdot) \approx P_s(\mu)$. Substituting the GM expression of the state (10) into (3) and (4), the single-state prediction parameters are given by:

$$r_{t|t-1}^n = r_{t-1}^n \sum_{i=1}^{N_{t-1}^n} \omega_t^{n,i} P_s(\boldsymbol{\mu}_{t|t-1}^{n,i}) \tag{12}$$

where the predicted Gaussian component is given by:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{t|t-1}^{n,i}, \boldsymbol{\Sigma}_{t|t-1}^{n,i}) = \langle f(\mathbf{x}|\cdot), \mathcal{N}(\cdot; \boldsymbol{\mu}_{t-1}^{n,i}, \boldsymbol{\Sigma}_{t-1}^{n,i}) \rangle \tag{13}$$

with predicted weights:

$$\omega_{t|t-1}^{n,i} = \frac{\omega_t^{n,i} P_s(\boldsymbol{\mu}_{t|t-1}^{n,i})}{\sum_{j=1}^{N_{t-1}^n} \omega_t^{n,j} P_s(\boldsymbol{\mu}_{t|t-1}^{n,j})}. \tag{14}$$

Replacing the PDFs from (5) with Gaussian functions, the cost matrix is approximated as follows:

$$\eta_n(m) = \begin{cases} 1 - r_{t|t-1}^n + r_{t|t-1}^n \sum_{i=1}^{N_{t-1}^n} \omega_{\text{miss}}^{n,i} & \text{if } m = 0 \\ \dfrac{r_{t|t-1}^n \sum_{i=1}^{N_{t-1}^n} \omega_{\text{dets}}^{n,i,m}}{\kappa(\mathbf{z}_m) + P_D(\boldsymbol{\mu}_m) D_{B,t}(\boldsymbol{\mu}_m)}, & \text{if } m \geq 1 \end{cases} \tag{15}$$

where:

$$\omega_{\text{miss}}^{n,i} = \omega_{t|t-1}^{n,i} \left(1 - P_D(\boldsymbol{\mu}_{t|t}^{n,i})\right), \tag{16}$$

$$\omega_{\text{dets}}^{n,i,m} = \omega_{t|t-1}^{n,i} q_{t|t}^{n,i}(\mathbf{z}_m) P_D(\boldsymbol{\mu}_{t|t}^{n,i}), \tag{17}$$

in which $\boldsymbol{\mu}_m$ is the projection of the measurement $\mathbf{z}_m$ into the state space, the intensity $D_{B,t}(\boldsymbol{\mu}_m) = N_{B,t}(\boldsymbol{\mu}_m)/A_B$ represents the number of targets $N_{B,t}(\boldsymbol{\mu}_m)$ born in the surveillance area A_B and $q_{t|t}^{n,i}(\mathbf{z}_m)$ is the measurement likelihood, which

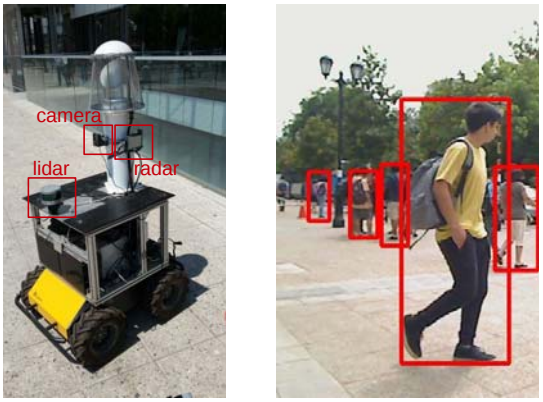

Figure 1. Left: The vehicle with the lidar, radar and visual sensors. Right: The detection of people (red rectangles) using faster R-CNN.

can be computed using a Kalman, EKF or UKF corrector. Linear expressions for the posterior parameters $q_{t|t}^{n,i}(\mathbf{z}_m)$, $\boldsymbol{\mu}_{t|t-1}^{n,i}$ and $\boldsymbol{\Sigma}_{t|t-1}^{n,i}$ can be found in [11]. The posterior Gaussian component is given by:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{t|t}^{n,i}, \boldsymbol{\Sigma}_{t|t}^{n,i}) = \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{t|t-1}^{n,i}, \boldsymbol{\Sigma}_{t|t-1}^{n,i}) f_z(\mathbf{z}_m|\mathbf{x})}{q_{t|t}^{n,i}(\mathbf{z}_m)}. \tag{18}$$

The posterior probability of existence for the detected targets is given by:

$$r_{t|t}^n = \frac{h_{n,0} r_{t|t-1}^n \sum_{i=1}^{N_{t-1}^n} \omega_{\text{miss}}^{n,i}}{1 - r_{t|t-1}^n + r_{t|t-1}^n \sum_{i=1}^{N_{t-1}^n} \omega_{\text{miss}}^{n,i}} + \sum_{m=1}^{M} h_{n,m}, \tag{19}$$

and for the birth targets is given by:

$$r_{t|t}^{N+m} = \frac{h_{0,m} P_D(\boldsymbol{\mu}_m) D_{B,t}(\boldsymbol{\mu}_m)}{\kappa(\mathbf{z}_m) + P_D(\boldsymbol{\mu}_m) D_{B,t}(\boldsymbol{\mu}_m)}, \tag{20}$$

and the corresponding weights are given by:

$$w_{t|t}^{n,i} = h_{n,0} \omega_{t|t-1}^{n,i}, \tag{21}$$

$$w_{t|t}^{n,N_{t-1}^n m+i} = h_{n,m} \frac{\omega_{\text{dets}}^{n,i,m}}{\sum_{j=1}^{N_{t-1}^n} \omega_{\text{dets}}^{n,j,m}}. \tag{22}$$

The new born hypothesis is composed of a single Gaussian, thus $\omega^{N+m} = 1$.

As can be seen in (21) and (22) the posterior GM increases its number of components from N_n to $N_n(1+M)$. Therefore, pruning of the Gaussian components must be carried out after each time step in order to maintain the tractability of the filter. Closely spaced Gaussian functions are merged and those with low weights removed as in [12].

IV. Experiment Description

The experiment consisted of people walking in an urban environment and three sensors recording the scene from a fixed position. The sensors were a Velodyne VLP-16 lidar, a Delphi SRR2 radar and a generic USB camera - see Figure 1 (left). The radar internally processes detections, with an allocated space for up to 64 detections. It has a field of view of up

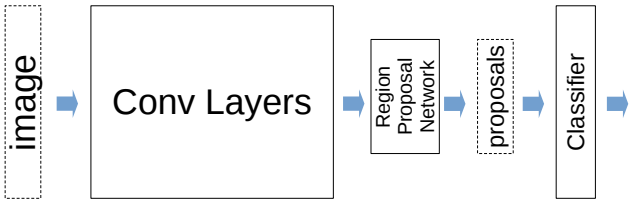

Figure 2. Faster R-CNN network for multi-class object detection.

to $80m$ in range and $\pm 75^\circ$ in bearing. The detections consist of position r_r (range), θ_r (angle), $\dot{r}_r$ (radial velocity) and the amplitude of the received signal in dBsm (decibels relative to one square meter). The frequency of measurements of the radar is 20 fps. The lidar creates 3D images by using 16 individual lasers (channels), each channel scanning through 360° in bearing. Each channel is separated from the next by an elevation angle of 2°, in the interval $[-15^\circ, 15^\circ]$. The camera has a resolution of 640×480 pixels, with a frame rate of 30 fps, and the distortion parameters, projection and camera calibration matrix were obtained using a calibration methodology available in the Robotic Operating System (ROS) [13], [14].

A. Detections

In order to use the GM-MBP algorithm (and the δ-GMBP [15] and δ-GLMB algorithms for comparison purposes), target detections are needed. Because the mono-pulse radar already detects targets, no other detector is required. However, the lidar and camera need detection methods. The detector used by the lidar was developed in [5], which is based on clustering using the known average size and shape of people. For people detection in images from the camera, a deep convolutional neural network was used [16]. For the radar and the lidar, background removal was carried out in order to reduce clutter and false alarms, the procedure for which is detailed in [5]. The procedure for detecting people in images from the camera is now explained.

People detection with the camera images: Convolutional neural networks have shown impressive results in object detection and classification in images, under variable sensor and environmental conditions [17], [18]. In order to detect people, the neural network "faster R-CNN" [19] was chosen because of its speed, and reported high quality results. Figure 2 shows the concept behind faster R-CNN, which produces target detections and their classifications. The images have distortions produced by the camera lens. In order to project from the world to image coordinates, it is necessary to undistort the images. Since faster R-CNN performs equally well with distorted or corrected images [13], the procedure can be of two forms. One way is to pass the corrected image to the detector. The other way is to carry out detections in the distorted image, and correct the coordinates of the detections.

The faster R-CNN detects 20 different object classes, such as people, vehicles, bicycles, motorcycles, airplanes, etc. Each detection performs a classification of all the 20 classes, returning a score that represents the probability of belonging to a class. This score measures the relation between the object and the background, i.e., the Signal to Noise Ratio (SNR) of each class. In this experiment the interest is in detecting people and for this reason a valid detection corresponds to the case in which the highest score corresponds to the class "person".

An example of the detection of people can be seen in Figure 1 (right), which shows five detected people (red rectangles).

B. Tracking

The lidar, camera and radar are assumed to provide conditionally independent measurements with respect to the target state, and each measures at its own frame rate. The δ-GMBP filter obtains its data from each sensor at any time, since the data is not synchronized. The prediction of the state using the kinematic model is therefore computed using the increment of time since the arrival of the previous measurement, no matter which sensor the data is from. The state is then corrected using the observation model corresponding to the current sensor.

The use of different sensors with different noise sources should not affect the target state estimates, because the sensor observation models take into account the different statistics of each sensor.

1) Birth Model: The multi-sensor system generates samples with irregular periods. N_B new targets are expected to be born every ΔT seconds uniformly spaced in an area A_B, however, if the periods are irregular, the number of new targets should be proportional to the time step between sample $t-1$ and t - i.e., ΔT_t. Let $D_B = N_B/A_B$ represent the intensity of the Poisson RFS, for the expected period ΔT. Thus, the intensity of the Poisson RFS in t should be given by $D_{B,t} = D_B \frac{\Delta T_t}{\Delta T}$.

New targets (people) first appear at the borders of the sensors' fields of view. For this reason, the birth intensity should be assigned a high value at the borders, and a lower value in the interior, as represented in Figure 3a. In this way the birth model used in the experiments with the MBP filter is uniform in each region, with a high value at the borders and a lower value in the interior.

2) Probability of Detection: A unique probability of detection is assigned to each sensor and is assumed to be independent of time and constant.

3) Probability of Survival: The probability of survival is modelled as being dependent on the:

- variable sampling rate,
- location of the state,
- time since the target state was born,

all of which will now be explained.

Variable sampling rate: In a manner similar to the target birth intensity function, the target survival probability must be adjusted according to the variable time period between samples. Let the probability of survival for a constant period ΔT be $P_s(\Delta T)$. Let $\nu = \Delta T/\Delta T_t$ be the number of samples in the period ΔT. Then $P_s(\Delta T) = (P_{s,t})^\nu = (P_{s,t})^{\Delta T/\Delta T_t}$.

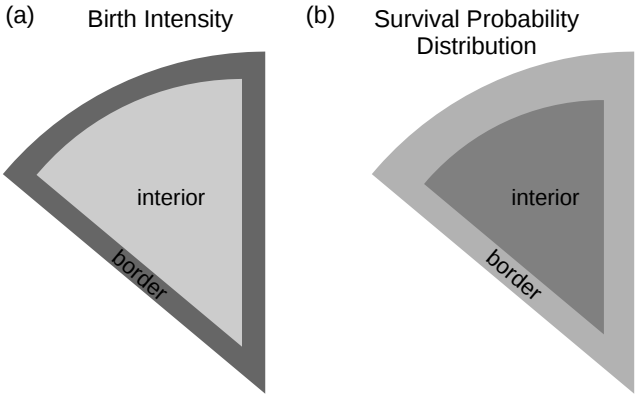

Figure 3. (a) Uniform birth intensity regions and (b) survival probability distribution. The darker shades represent higher values.

Therefore, solving for $P_{s,t}$, gives the probability of survival at the current time step:

$$P_{s,t} = \exp\left(\frac{\Delta T_t}{\Delta T}\log(P_s(\Delta T))\right). \tag{23}$$

Location of the state: In contrast to target births, a target is expected to persist in the interior section of the common field of view of the sensors and disappear at the borders when the target exits. This is shown in Figure 3b.

Time since the target state was born: In [20] Kim and Vo incorporated the time of persistence of the tracks to compute the probability of survival in order to delay early track termination due to occlusions. Since the state's label l contains the time of birth of the target, the time persistence is the difference between the current time and that contained in the label l. The probability of survival varies in time starting at value $P_{s,0}$ when the target is born, to $P_{s,\tau}$ after τ seconds. A hyperbolic tangent function is therefore used to model the survival probability $P_{s,t}(l)$

$$P_{s,t}(l) = P_{s,0} + (P_{s,\tau} - P_{s,0})\tanh\left(\frac{\Delta T_{t,l}}{\tau}\exp(1)\right) \tag{24}$$

where $\Delta T_{t,l}$ is the time of existence of the target - i.e., the difference between the current time and that encoded in label l. The hyperbolic tangent allows the fast and asymptotic change from $P_{s,0}$ to $P_{s,\tau}$.

4) Target State Transition Model: The kinematic state of a person is given by a vector of positions and velocities in the ground plane $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$. The state transition function is assumed to be linear with constant velocity and covariance matrix Q as follows:

$$\begin{aligned}\begin{bmatrix} x_{t|t-1} \\ y_{t|t-1} \\ \dot{x}_{t|t-1} \\ \dot{y}_{t|t-1} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & \Delta T_t & 0 \\ 0 & 1 & 0 & \Delta T_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} \\ \mathbf{Q}_t &= \sigma^2 \begin{bmatrix} {\Delta T_t}^2/2 & 0 \\ 0 & {\Delta T_t}^2/2 \\ \Delta T_t & 0 \\ 0 & \Delta T_t \end{bmatrix} \begin{bmatrix} {\Delta T_t}^2/2 & 0 \\ 0 & {\Delta T_t}^2/2 \\ \Delta T_t & 0 \\ 0 & \Delta T_t \end{bmatrix}^T \end{aligned} \tag{25}$$

where $\mathbf{Q}_t$ is the covariance matrix associated with the motion transition process, and σ corresponds to the acceleration standard deviation.

5) Measurement Model of the Mono-pulse Radar: In order to use the GM-MBP filter with the radar, a likelihood function that relates a measurement with the state of a track must be designed. The measurement vector is $\mathbf{z}_r = [r_r, \theta_r, \dot{r}_r]^T$.

Due to the non-linearity of the measurement in relation to the state, the Unscented Transform (UT) [21] is used to map the statistics of the measurement and states. An observation likelihood function $\mathbf{z}_r = g_{z_r}(\mathbf{x})$ must be determined, in order to correct the prediction made by the state transition model.

Radar measurements and target states have different coordinate systems. Small errors in the angular rotations between both coordinate systems produce large errors related to the measurements of targets located far from the sensor. For this reason a 3D rotation and translation transformation must be included, even when the sensor measures in a 2D plane. The relation between both, radar and state coordinate systems is given by the typical rotation and translation relation

$$\mathbf{x}^{3\mathrm{D}} = \mathbf{R}\cdot\mathbf{z}_r^{3\mathrm{D}} + \mathbf{t}_r^{3\mathrm{D}}, \tag{26}$$

in which $\mathbf{x}^{3\mathrm{D}} = [x, y, z]^T$ and $\mathbf{z}_r^{3\mathrm{D}} = [x_r, y_r, z_r]^T$ represent the 3D Cartesian positions in the state and radar coordinate systems respectively, and $\mathbf{t}_r^{3\mathrm{D}} = [t_{x_r}, t_{y_r}, t_{z_r}]^T$ represents the translation vector between the radar and state origin, while $\mathbf{R} = [r_{ij}]$, $i \in \{1,2,3\}$ and $j \in \{1,2,3\}$, is the 3D rotation matrix.

The value of z is not known, however, $z_r = 0$ because the radar is assumed to measure in a plane and does not have a vertical component. Therefore z can be computed from the state using the third row of:

$$\mathbf{z}_r^{3\mathrm{D}} = \mathbf{R}^T\left(\mathbf{x}^{3\mathrm{D}} - \mathbf{t}_r^{3\mathrm{D}}\right), \tag{27}$$

where (27) is the inverse solution of (26) and $R^T = R^{-1}$, resulting in:

$$z - t_{z_r} = -\frac{1}{r_{33}}\begin{bmatrix} r_{13} & r_{23} \end{bmatrix}\begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \end{bmatrix}. \tag{28}$$

Rewriting Equation (27), and substituting (28) in (27):

$$\begin{aligned}\begin{bmatrix} x_r \\ y_r \end{bmatrix} &= \mathbf{R}_{2\times 2}^T\begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \end{bmatrix} + \begin{bmatrix} r_{31}(z - t_{z_r}) \\ r_{32}(z - t_{z_r}) \end{bmatrix} \\ &= \left[\mathbf{R}_{2\times 2}^T - \frac{1}{r_{33}}\begin{bmatrix} r_{13}r_{31} & r_{23}r_{31} \\ r_{13}r_{32} & r_{23}r_{32} \end{bmatrix}\right]\begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \end{bmatrix} \\ &= \tilde{\mathbf{R}}^{-1}\begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \end{bmatrix} \end{aligned} \tag{29}$$

where $\mathbf{R}_{2\times2}$ corresponds to the sub-matrix formed by the first two rows and columns of $\mathbf{R}$. The radar also measures radial velocity, thus, we need to include the velocity component in the single-state likelihood distribution. By differentiating (29) with respect to time, the velocity components of the state are obtained. The transformation between state and measurements in Cartesian coordinates is then given by:

$$\begin{bmatrix} x_r \\ y_r \\ \dot{x}_r \\ \dot{y}_r \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}^{-1} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \tilde{\mathbf{R}}^{-1} \end{bmatrix} \begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \\ \dot{x} \\ \dot{y} \end{bmatrix}, \tag{30}$$

where $\mathbf{0}_{2\times2}$ is a matrix of zeros of dimension 2×2.

As the radar also computes radial velocity, the relation with the velocity components of the state must be included. The corresponding position and velocity relations are:

$$\begin{aligned} r_r &= \sqrt{x_r^2 + y_r^2} \\ \theta_r &= \arctan(y_r/x_r) \\ \dot{r}_r &= \dot{x}_r \cos\theta_r + \dot{y}_r \sin\theta_r. \end{aligned} \tag{31}$$

6) Measurement Model of the Velodyne Lidar: The detections of people obtained from the lidar are similar to the radar, but do not include a velocity component. The measurement vector is $\mathbf{z}_l = [r_l, \theta_l]^T$, which corresponds to the lidar origin to target detection range and bearing values. The likelihood function for the lidar is also very similar to the radar. The relationship between the observation and state vectors is:

$$\begin{aligned} r_l &= \sqrt{x_l^2 + y_l^2} \\ \theta_l &= \arctan(y_l/x_l) \end{aligned} \tag{32}$$

where $\mathbf{x}^{3\mathrm{D}} = \mathbf{R} \cdot \mathbf{z}_l^{3\mathrm{D}} + \mathbf{t}_l$, in which $\mathbf{x}^{3\mathrm{D}} = [x, y, z]^T$, $\mathbf{z}_l^{3\mathrm{D}} = [x_l, y_l, z_l]^T$ and $\mathbf{t}_l = [t_{x_l}, t_{y_l}, t_{z_l}]^T$ similarly to the radar in Section IV-B5.

7) Measurement Model of the Camera: The 4D measurement vector $\mathbf{z}_c^{4D}$ is represented by the coordinates of the center and the width and height of the detection, as seen in the rectangles in Figure 1 (right). In order to project the state $\mathbf{x}$ into $\mathbf{z}_c^{4D}$, the person is modeled as a cylinder, the surface points of which are projected into the 2D-image using the information obtained from the camera calibration.

The projection matrix $\mathbf{P}$ relates the pixel and real world coordinates:

$$\mathbf{P} = \begin{bmatrix} \mathbf{C} & \mathbf{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{33}$$

where α_u and α_v are the focal distances (relative to both image plane axes) in pixels, and u_0 and v_0 the coordinates representing the center of the camera, relative to the image plane origin. The axis rotation transformation matrix

$$\mathbf{H}_v = \begin{bmatrix} \mathbf{R}_v & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{34}$$

rotates the image axes to the camera coordinate axes, in which $\mathbf{R}_v$ is the rotation matrix containing the yaw and roll angles, which are both equal to $-\pi/2$.

The location and rotation of the camera coordinate system with respect to the state coordinate system, are given in its homogeneous form:

$$\mathbf{H}_b = \begin{bmatrix} \mathbf{R}_b & \mathbf{t}_b \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}. \tag{35}$$

in which $\mathbf{R}_b$ is the rotation matrix and $\mathbf{t}_b = [t_{x_b}, t_{y_b}, t_{z_b}]^T$ is the translation vector between the camera's origin and the global state origin. The rotation matrices are computed using the Euler angles [22].

The relation between the camera coordinates in pixels $\mathbf{z}_c^{2D} = [x_c, y_c]^T$ and the state coordinates $\mathbf{z}^{3D}$ is given by:

$$s \begin{bmatrix} \mathbf{z}_c^{2D} \\ 1 \end{bmatrix} = \mathbf{P}\mathbf{H}_v^{-1}\mathbf{H}_b^{-1}\mathbf{x}^{3\mathrm{D}} = \mathbf{C}\mathbf{R}_v^T\mathbf{R}_b^T \left(\mathbf{x}^{3\mathrm{D}} - \mathbf{t}_b\right) \tag{36}$$

where s is the scale factor of the 3D to 2D projection. Let $\mathbf{M} = \mathbf{C}\mathbf{R}_v^T\mathbf{R}_b^T$. The scale factor s can be computed solving the third row of (36), resulting in (37).

$$s = \mathbf{M}_{(3,:)} \left(\mathbf{x}^{3\mathrm{D}} - \mathbf{t}_b\right) \tag{37}$$

where $\mathbf{M}_{(3,:)}$ is the third row of $\mathbf{M}$. Then, solving the first two rows of (36) the camera measurement model is given by:

$$\mathbf{z}_c^{2D} = s^{-1}\mathbf{M}_{(1:2,:)} \left(\mathbf{x}^{3\mathrm{D}} - \mathbf{t}_b\right) \tag{38}$$

where $\mathbf{M}_{(1:2,:)}$ is the submatrix of $\mathbf{M}$ (its first two rows).

The cylinder, centered at the ground level positional component of the state $\mathbf{x}$, is quantized and its points are projected into the image using Equation (38) multiple times, generating a set of projected points Z_c^{2D}. The rectangle $\mathbf{z}_c^{4D}$ representing the projected target is then obtained from the minimum and maximum coordinates in Z_c^{2D}.

V. Results

All of the filter results are based on GM implementations, and therefore the GM abbreviation will be omitted from here on. Figure 4 shows a sequence of three images of detections and track estimates using the MBP filter. In the figure, the common intersection between the fields of view of the sensors was used for performing the experiments.

A. Performance Comparison of the MBP, δ-GMBP and δ-GLMB Filters with a Single Sensor

The δ-GLMB filter is one of the most accepted recent RFS based multi-target tracking algorithms. For this reason, the δ-GLMB filter is used in this experiment, as a benchmark comparison for the MBP filter. We also compare results with the δ-GMBP filter, because it is very similar to the δ-GLMB filter, except for its birth model. Also, the MBP filter is an approximation of the δ-GMBP filter, in a manner analogous to the LMB filter being an approximation of the δ-GLMB filter. For comparison purposes all filters are implemented with identical detection statistical parameters. It should be noted that, in the δ-GLMB and δ-GMBP filter implementations

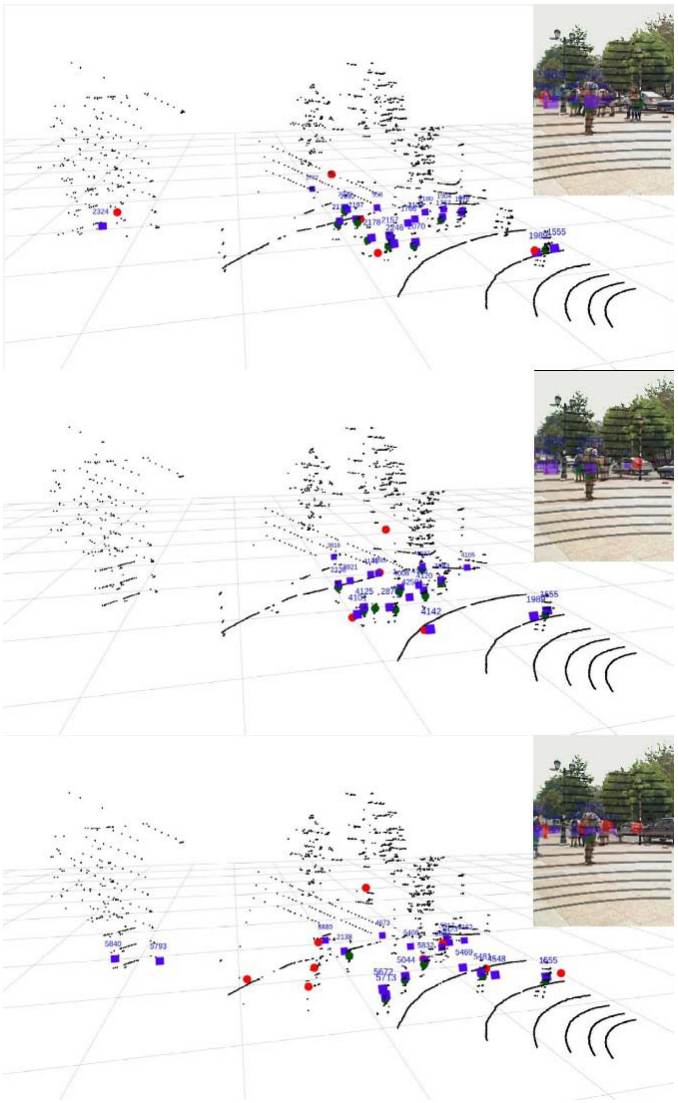

Figure 4. A sequence of three images based on the MBP filter. Green circles represent the lidar detections, red circles the radar detections, blue squares the MBP filter estimates and black points the raw lidar measurements. In the upper right side of each image a camera view is seen with the detections and estimates superimposed. The numbers correspond to the unique labels assigned to estimated target tracks.

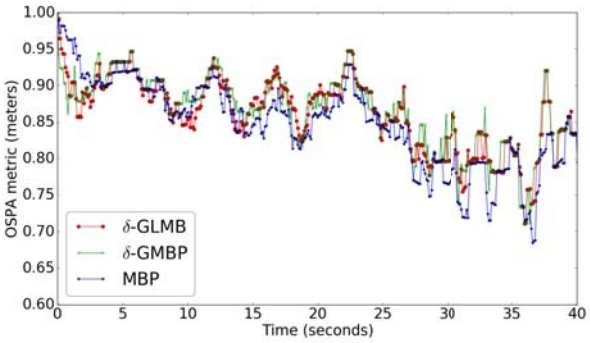

Figure 5. The OSPA tracking error metric using only the lidar. The OSPA metric parameters used throughout this article where cut-off $c = 1.0$m and power $p = 2$.

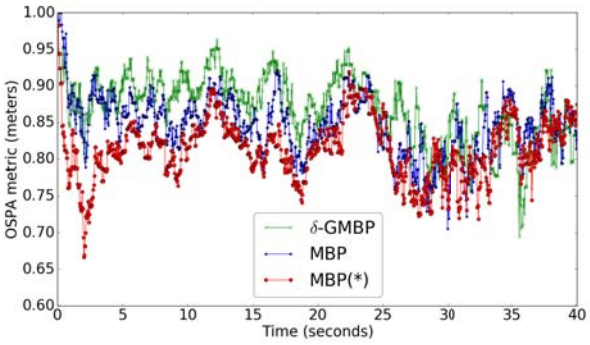

Figure 6. Comparison between the proposed MBP and δ-GMBP filters in [5]. The green curve shows the OSPA error for the δ-GMBP filter, the blue curve that for the MBP filter with standard statistics and the red curve the OSPA error for the MBP filter with improved statistics (with (*) symbol at the legend).

currently available to the authors, the updates are implemented with Murty's algorithm. However, the MBP filter adopts the proposed multi-target prediction and update histogram based, Gibbs sampling approach.

Initial results of the δ-GLMB, δ-GMBP and MBP filters, based only on lidar measurements, are shown in Figure 5. In this figure, the Optimal Sub-Pattern Assignment (OSPA) metric is used to assess each filters' performance [23]. In all experiments, the OSPA cut-off parameter c was set to 1.0m and the power p to 2. The figure shows similar results for the three filters. Despite the loss of correlation information between the targets and measurements, their similar performances can be explained due to the fact that the MBP filter does not need to truncate many components, which represent the posterior distribution. This is in contrast to the δ-GLMB and the δ-GMBP filters, which require more parameters to represent their multi-target distributions. Since the δ-GLMB and the δ-GMBP filter updates are implemented with Murty's algorithm, instead of Gibbs sampling, in order to run the filters in a reasonable time the truncation of the multi-Bernoulli mixture components was necessary. This negatively impacts their estimation accuracy.

B. Performance Comparison of the MBP and δ-GMBP Filters with the Radar and Lidar

As shown in Figure 6, both the MBP and the δ-GMBP filters are capable of multi-target tracking, based on measurements from two sensors (the radar and lidar). Similarly to the results of the previous section, the MBP filter without the performance enhancing changes of Section IV-B, performs slightly better than the δ-GMBP filter. However, the MBP filter, with the enhancements of Section IV-B, performs consistently better, displaying lower OSPA errors at most time steps.

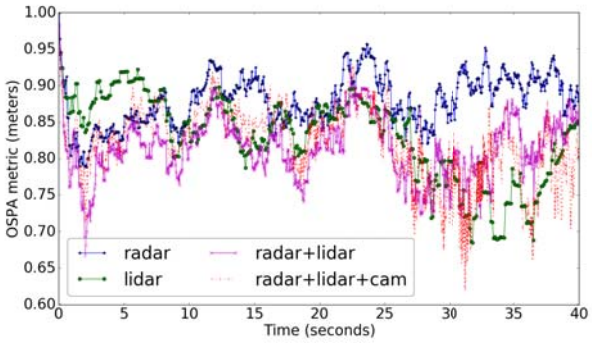

Figure 7. The OSPA metric performance values of the MBP filter, using different sensor combinations. The blue curve shows the radar only, the green curve the lidar only, the magenta curve shows the fusion of lidar and radar, and red curve shows the fusion of lidar, radar and camera.

C. Performance Comparison of the MBP filter with Differing Sensor Combinations

Figure 7 shows the performance of the MBP filter with different sensor combinations. As expected, the track estimates based on the fusion of radar and lidar are superior to those when only the single sensors (lidar or radar) are used. Importantly, when using both the lidar and radar, the MBP filter is capable of maintaining the tracks when individual sensors perform poorly. This is particularly evident from time 0 to 10s, when lidar only based tracking has large OSPA errors and from time 28s to 38s, when radar based tracking yields high OSPA errors.

It is important to note that the MBP filter based only on the camera is unable to track people, due to the lack of range information. However, when all three sensors are used, note that the image data contributes with very precise object detection, and a low false detection rate.

VI. Conclusions

The GM implementation of the MBP filter was presented in this article. It was evaluated in a multi-target multi-sensor scenario, using measurements from a lidar, radar and camera.

In the scenario tested, comparisons with the state of the art δ-GLMB and δ-GMBP filters, demonstrated similar performance with lower computational times, despite the loss of target to measurement correlation information. This can be partly explained by the fewer parameters needed by the MBP filter to represent its multi-target state, when compared to the δ-GLMB or δ-GMBP filters and partly due to the Gibbs sampler used in the MBP filter, as opposed to Murty's algorithm adopted in the δ-GLMB and δ-GMBP filters.

For the MBP filter, it was demonstrated that modelling the dependencies of the target survival probability and birth intensity on variable sampling times and spatial considerations within the sensor field of view, enhanced its performance.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0386. This work was also funded by the Advanced Mining Technology Center (AMTC) and Conicyt-Fondecyt project 1150930.

References

[1] Mahler, R.P.S., *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, Inc., 2014.

[2] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The Labeled Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[3] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *Signal Processing, IEEE Transactions on*, vol. 61, no. 13, pp. 3460–3475, 2013.

[4] J. Correa, M. Adams, and C. Perez, "A dirac delta mixture-based random finite set filter," in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, Oct 2015, pp. 231–238.

[5] L. Cament, M. Adams, J. Correa, and C. Perez, "The -generalized multi-bernoulli poisson filter in a multi-sensor application," in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, October 2017.

[6] S. Reuter, A. Danzer, M. Stübler, A. Scheel, and K. Granström, "A fast implementation of the labeled multi-bernoulli filter using gibbs sampling," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 765–772.

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

[8] B.-N. Vo, B.-T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1975–1987, 2017.

[9] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *ArXiv e-prints*, Mar. 2012.

[10] Á. F. García-Fernández, J. L. Williams, K. Granstrom, and L. Svensson, "Poisson multi-bernoulli mixture filter: direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

[11] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, Nov 2006.

[12] B.-N. Vo, W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, November 2006.

[13] J.-Y. Bouguet, "Matlab camera calibration toolbox," *Caltech Technical Report*, 2000.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[15] J. Correa and M. Adams, "Estimating detection statistics within a Bayes-closed multi-object filter," in *Information Fusion (FUSION), 2016 19th International Conference on*. IEEE, 2016, pp. 811–819.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[18] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[20] D. Y. Kim, B.-N. Vo, and B.-T. Vo, "Online visual multi-object tracking via labeled random finite set filtering," *arXiv preprint arXiv:1611.06011*, 2016.

[21] S. J. Julier, "The spherical simplex unscented transformation," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3. IEEE, 2003, pp. 2430–2434.

[22] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[23] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE transactions on signal processing*, vol. 56, no. 8, pp. 3447–3457, 2008.