The δ-Generalized Multi-Bernoulli Poisson Filter in a Multi-Sensor Application

1st Leonardo Cament*, 2nd Martin Adams†, 3rd Javier Correa‡ and 4th Claudio Perez§
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, §clperez@ing.uchile.cl

***Abstract*—This paper proposes a multi-target tracking strategy using a δ-Generalized Multi-Bernoulli Poisson (δ-GMBP) filter applied in a multi-sensor scenario. The δ-GMBP distribution is closed under the Chapman-Kolmogorov equation and Bayes rule, and also closed for a wide family of multi-target likelihood functions which allows implementations of different kinematic and measurement models.**

One difference between the δ-GMBP and the state of the art of multi-Bernoulli filters is that the birth process is modeled with a Poisson Random Finite Set (RFS), which can be more intuitive. Further, in order to obtain the posterior of the δ-GMBP filter recursion, it is not necessary to iterate over all the components of the prior mixture. The δ-GMBP filter, also maintains track labels in the multi-Bernoulli components, thus no other association method is necessary.

The experiments carried out consist of people walking in an open place and two sensors recording the scene from a fixed position. The sensors used in the experiment are a 3D lidar and a single-beam mono-pulse radar.

The δ-GMBP filter is compared with the classical Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter, and the Marginal Multi-target Multi-Bernoulli (m-MeMBer) filter.

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

I. Introduction

This paper proposes a multi-target tracking strategy using a δ-GMBP filter applied in a multi-sensor scenario. The δ-GMBP RFS is a combination of a δ-Generalized Multi-Bernoulli (δ-GMB) RFS and a Poisson RFS. The δ-GMB RFS models the known targets, similarly to the δ-Generalized Labeled Multi-Bernoulli (δ-GLMB) [1], with the difference that the potential targets that have not yet been detected are modeled as a Poisson RFS, as in the m-MeMBer filter [2]. The δ-GMBP distribution is closed under the Chapman-Kolmogorov equation and Bayes rule, and also closed for a wide family of multi-target likelihood functions which allows implementations of different kinematic and measurement models.

One difference between the δ-GMBP and the state of the art of multi-Bernoulli filters is that the birth process is modeled with a Poisson RFS, which can be more intuitive. A justification for this can be seen by assuming that the expected number of new born targets is known. If the new born targets appear following a Bernoulli process with equal probability of being born, the maximum entropy distribution modeling these assumptions is a Poisson distribution [3].

Funded by the Advanced Mining Technology Center (AMTC) and Conicyt-Fondecyt project 1150930.

A fundamental advantage of the δ-GMBP filter is that it maintains track labels in the multi-Bernoulli components, thus no other association method is necessary. Further, in order to obtain the posterior of the δ-GMBP filter recursion, a version of which was first presented in [4], it is not necessary to iterate over all the components of the prior mixture. The solution consists of an iterative algorithm which solves cost matrices in order to obtain components of the δ-GMBP RFS that contribute the highest weights, to the mixture. As carried out by Vo et al. in [1], the cost assignment is solved with Murty's algorithm, although for the δ-GLMB filter faster implementations using a Gibbs sampler [5], [6] have already been implemented in [7], which will be examined in future work for the proposed δ-GMBP filter. Despite the maintained track labels, this article will evaluate this algorithm based on an unlabeled OSPA metric [8] for comparison with unlabeled filters.

The experiment carried out consists of people walking in an open place and two sensors recording the scene from fixed positions. The sensors used in the experiment are a 3D lidar and a single-beam mono-pulse radar. The sensors are assumed independent and, thus measurements are conditionally independent of the target states. When a new measurement is received by the system, no matter which sensor it is from, an update and correction using the filter is made, i.e., in an asynchronous form. This is carried out because in real applications, the data is extracted by different sensors at different times. Therefore any component created due to a sensor measurement can only be eliminated after all sensor measurements are received. This avoids the rapid elimination of components in the updated multi-Bernoulli mixture.

The δ-GMBP filter is compared with the GM-PHD filter [9], a Gaussian Mixture implementation of Mahler's classical Probability Hypothesis Density (PHD) filter [10], and Williams' m-MeMBer filter [2]. This is a MeMBer filter [10] that models target births with a Poisson RFS. The m-MeMBer filter was implemented using Murty's algorithm [11], as used in [1], instead of the Loopy Belief Propagation (LBP) algorithm [12] used in [2], to ensure convergence.

II. Theoretical Background

This section briefly describes the theory behind the δ-GMBP filter. A complete description of the filter is given in [13].

A. Random Finite Sets Overview

A 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 in the set, 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 (1), while the correction uses Bayes rule (2)

$$p(X_{t+1}|Z_{1:t}) = \int p(X_{t+1}|X_t)p(X_t|Z_{1:t})\delta X_t \qquad (1)$$

$$p(X_{t+1}|Z_{1:t+1}) \propto p(Z_{t+1}|X_{t+1})p(X_{t+1}|Z_{1:t}), \qquad (2)$$

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

In order to solve Equations 1 and 2 the transition and measurement models must fulfill various properties [10, p. 313].

C. The δ-Generalized Multi-Bernoulli Poisson Filter

The δ-GMBP RFS is a combination of a δ-GMB RFS and a Poisson RFS which is closed under the prediction and correction steps. The δ-GMB RFS models the known targets, while the Poisson RFS models the potential targets that have not yet been detected:

$$X \sim \underbrace{\text{δ-Generalized Multi-Bernoulli}(\boldsymbol{\Theta}) \cup \text{Poisson}(D_{\mathcal{U}})}_{\text{δ-GMBP}(\boldsymbol{\Theta}, D_{\mathcal{U}})} \qquad (3)$$

where $D_{\mathcal{U}}$ represents the density of a Poisson RFS and $\boldsymbol{\Theta}$ is a set with all the parameters of the δ-GMB RFS.

A cost matrix containing the logarithm of the weights of the components of the posterior δ-GMBP distribution is built using the prior δ-GMBP components and the measurements. The cost matrix has a physical interpretation. It explicitly includes cost values, which correspond to clutter, new targets and miss detections as well as known detections, surviving and non-surviving targets. The cost assignment is solved using Murty's algorithm [11]. The details and a pseudo-code implementation of the recursion, prediction and correction can be found in [4], [13].

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 he inverse likelihood function $\mathbf{x} \sim g_z^{-1}(\mathbf{z}, \cdot)$.

To extract the estimated state from the filter, the first moment of the Generalized Multi-Bernoulli RFS is used. When using a Gaussian mixture density implementation of the transition and observation models, the target states are obtained from the peaks of the first moment of the δ-GMBP distribution. In the experiments here, peaks, with weights greater than 0.5 are assumed to be targets.

III. Experiment

The experiment consisted of people walking in an open place and two sensors recording the scene from a fixed position. The sensors were a Velodyne VLP-16 lidar and a Delphi RSDS radar. The radar internally processes detections, with an allocated space for up to 64 detections. It has a field of view of up to 200 m in range and $\pm 100^\circ$ in bearing. The detections consist of position r (range), θ (angle), $\dot{r}$ (radial velocity) and the amplitude of the 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 $[-15^\circ, 15^\circ]$.

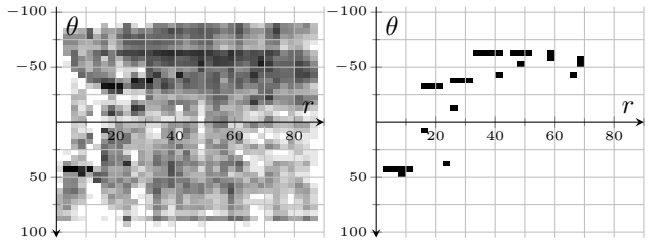
A. Detections

In order to use the δ-GMBP algorithm [13], detections are needed. As the mono-pulse radar already detects targets, no other detector is required. However, for both the radar and lidar, background removal is carried out in order to reduce clutter and false alarms. This procedure is now explained.

1) Radar and Lidar Background Removal: Background estimation is carried out by calculating a two dimensional histogram of the reported radar detections or raw lidar points. In the case of the lidar, the 3D positions of the raw points are projected onto a 2D plane by removing the z coordinate. First, the detection area is discretized into polar bins, which are indexed by their bearing angle bin number θ_i, and range bin number r_j. For each sample, the bin corresponding to a detection is computed and the histogram bin is incremented by 1. The resulting histogram is used to model the background measurements.

Figure 1 shows the radar background detection results. In the left the histogram, shown with a logarithmic intensity scale, darker shades indicate a higher number of repeated positions detected by the radar. The right graph shows the estimated background points in black after applying a threshold to the histogram.

2) People Detection with the Lidar: The implemented lidar detector only detects people between coordinates $z = 0$ and $2m$, based on the assumption that people are all on the ground plane and that their heights do not exceed $2m$. The detection



(a) Overlap of radar detections in polar coordinates.

(b) Background estimation in polar coordinates.

Figure 1. The computed histogram for background removal for the monopulse radar. The left graph shows the histogram with a logarithmic intensity scale and right graph shows the background after using a threshold on the histogram. Darker colors indicate a higher number of repeated positions detected by the radar.

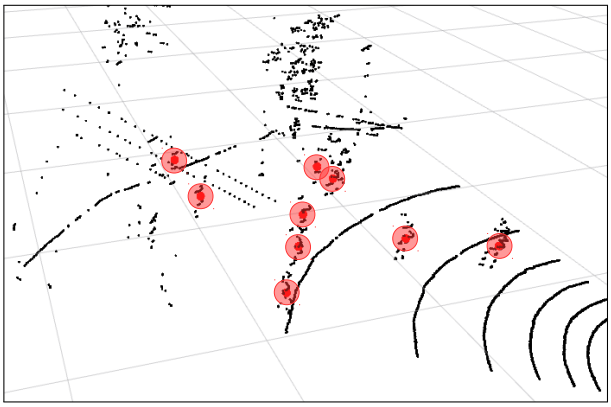


Figure 2. The detection of people using the Velodyne lidar. The small black points represent the 3D lidar data and red circles represent the detections.

is performed in two steps. Firstly, clustering by channel takes place, in which each channel corresponds to a laser angle. Secondly clustering the detections among all the channels is then carried out.

Clustering by channel: The data per channel is a 3D vector of (x, y, z) coordinates, sorted by bearing angle in an ascending order. Two consecutive points of a j-th channel are assumed to belong to the same object if the distance between them is smaller than a given threshold. The process produces N sets of points $P_1^j, ..., P_N^j$, that may contain all, or a fraction of, points belonging to the object. DBSCAN clustering [14] is used to join those sets P_k^j that were not joined in the previous process, thus producing new sets of points $\tilde{P}_1^j, ..., \tilde{P}_{\tilde{N}}^j$. DBSCAN allows the use of a distance matrix to perform the clustering operation. Therefore, it is possible to cluster general structures, which are represented as sets (P_k in this case), in contrast to most state of the art clustering methods, which can only group vectors. The distance matrix $D_j(k, l)$ represents the distance between P_k^j and P_l^j. The distance $D_j(k, l)$ corresponds to the combined perimeter formed by P_k^j and P_l^j. When the combined perimeter is larger than $2m$, P_k^j and P_l^j are considered to be part of different objects, thus, the distance value is penalized with a high value ($D_j(k, l) = 1000$).

Clustering among all channels: The elements to be grouped are new sets of points $\tilde{P}_1^j, ..., \tilde{P}_{\tilde{N}}^j$, that are reorganized as $\tilde{P}_1, ..., \tilde{P}_{\tilde{M}}$, where $\tilde{M} = \sum_{j=0}^{15} \tilde{N}^j$. Clustering is then again carried out using the DBSCAN algorithm, and the distance corresponds to:

$$\hat{D}(\tilde{P}_k, \tilde{P}_l) = \left(1 - \frac{A(\tilde{P}_k) \cap A(\tilde{P}_l)}{A(\tilde{P}_k) \cup A(\tilde{P}_l)}\right) \cdot \frac{A(\tilde{P}_k \cup \tilde{P}_l)}{A(\tilde{P}_k) \cup A(\tilde{P}_l)} \quad (4)$$

where the first factor is the Jaccard dissimilarity, multiplied by the ratio between the convex polygon generated by the points of $\tilde{P}_k \cup \tilde{P}_l$ and the union of the areas. When $\tilde{P}_k$ and $\tilde{P}_l$ do not intersect, the Jaccard dissimilarity is 1. However, it is necessary to measure how separated the areas in that scenario are, in order to join areas separated by a small distance. This is why the second factor is used. The distance is penalized with a high value (1000) to reject distant sets of points.

An example of people detection using the lidar can be seen in Figure 2.

B. Tracking

The δ-GMBP algorithm described in [13] is used to track people using the radar and the lidar.

All sensors are assumed to provide conditionally independent measurements, and each one measures with its own frame rate. The δ-GMBP filter obtains its data from both sensors 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 state target estimates, because the sensor observation models take into account the different statistics of each the sensor.

1) Transition Model: The kinematic state of a person is given by a vector of positions and velocities in the ground plane (x, frontal, and y lateral coordinates) $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$. The state transition function is assumed to be linear with constant velocity and covariance matrix Q, and follows Equation 5.

$$\begin{bmatrix} x_{k+1|k} \\ y_{k+1|k} \\ \dot{x}_{k+1|k} \\ \dot{y}_{k+1|k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{k|k} \\ y_{k|k} \\ \dot{x}_{k|k} \\ \dot{y}_{k|k} \end{bmatrix} + \varepsilon_{\Delta T}$$
$$Q = \sigma^2 \cdot \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \cdot \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}^T \quad (5)$$

where ΔT is the time step between consecutive samples, $\varepsilon_{\Delta T}$ is a zero mean transition noise vector with covariance Q, and σ corresponds to the acceleration standard deviation.

2) Measurement Model of the Mono-pulse Radar: In order to use the δ-GMBP algorithm with the radar, a likelihood function that relates a measurement with the state of a track must be designed. The measurement vector is $\mathbf{z}_r = [r, \theta, \dot{r}]^T$ which corresponds to the range, angle and range rate measured by the radar.

Due to the non-linearity of the measurement in relation to the state, the Unscented Transform (UT) [15] 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 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 $\vec{x} = R \cdot \vec{z}_r + t_r$, in which $\vec{x} = [x, y, z]^T$ and $\vec{z}_r = [x_r, y_r, z_r]^T$ represent the 3D Cartesian positions in the state and radar coordinate systems respectively, and $t_r = [t_{x_r}, t_{y_r}, t_{z_r}]^T$ represents the translation vector while $R = [r_{ij}]$, $i \in \{1, 2, 3\}$ and $j \in \{1, 2, 3\}$ is the 3D rotation matrix.

The value of z is not known, but it is known that $z_r = 0$ because the radar measures in a plane and does not have a vertical component. Therefore z can be computed from the state using the equation formed by the third equation of $\vec{x} = R \cdot \vec{z}_r + t_r$:

$$z - t_{z_r} = -\frac{1}{r_{33}} \cdot \begin{bmatrix} r_{13} & r_{23} \end{bmatrix} \cdot \begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \end{bmatrix}. \tag{6}$$

By differentiating 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:

$$\begin{bmatrix} x_r \\ y_r \\ \dot{x}_r \\ \dot{y}_r \end{bmatrix} = \begin{bmatrix} \tilde{R}^{-1} & 0_{2\times2} \\ 0_{2\times2} & \tilde{R}^{-1} \end{bmatrix} \cdot \begin{bmatrix} x - t_{x_r} \\ y - t_{y_r} \\ \dot{x} \\ \dot{y} \end{bmatrix}, \tag{7}$$

$$\text{where } \tilde{R}^{-1} = \begin{bmatrix} R^T_{2\times2} - \frac{1}{r_{33}} \cdot \begin{bmatrix} r_{13}r_{31} & r_{23}r_{31} \\ r_{13}r_{32} & r_{23}r_{32} \end{bmatrix} \end{bmatrix},$$

$R_{2\times2}$ corresponds to the sub-matrix formed by the first two rows and columns of R, and $0_{2\times2}$ is a matrix of zeros of the same dimension.

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 shown in Equations 8.

$$\begin{aligned} r &= \sqrt{x_r^2 + y_r^2} \\ \theta &= \arctan(y_r / x_r) \\ \dot{r} &= \dot{x}_r \cos\theta_r + \dot{y}_r \sin\theta. \end{aligned} \tag{8}$$

New born targets are obtained by sampling from the inverse measurement function $\tilde{\mathbf{x}} \sim g^{-1}_{z_r}(\mathbf{z}_r)$. The transformation for computing the new born targets is given by inverting 7 and 8.

3) 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 $\mathbf{z}_l = [r, \theta]$, which corresponds to range and bearing. The likelihood function for the lidar is also very similar to the radar. The relationship between the observation and state vectors follows Equations 9.

$$\begin{aligned} r &= \sqrt{x_l^2 + y_l^2} \\ \theta &= \arctan(y_l / x_l) \end{aligned} \tag{9}$$

where $\vec{x} = R \cdot \vec{z}_l + t_l$, in which $\vec{x} = [x, y, z]^T$, $\vec{z}_l = [x_l, y_l, z_l]^T$ and $t_l = [t_{x_l}, t_{y_l}, t_{z_l}]^T$ as with the radar in Section III-B2.

New born targets are sampled in the same way from the inverse measurement function $\tilde{\mathbf{x}} \sim g^{-1}_{z_l}(\mathbf{z}_l)$.

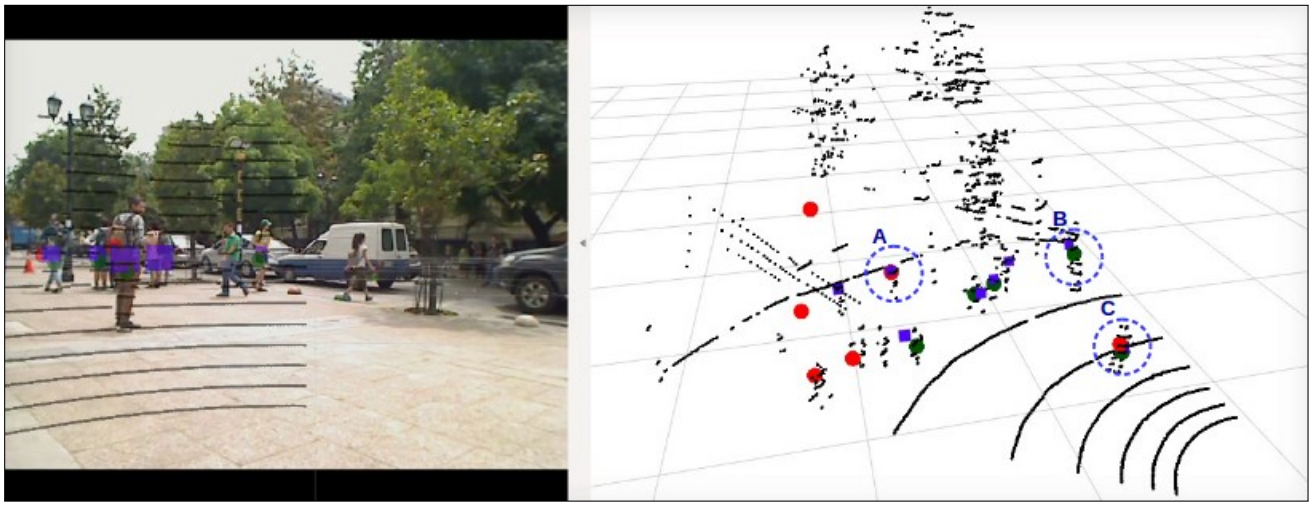
IV. Results

For evaluation purposes here, some images from the results using the δ-GMBP filter are shown in Figure 3. The right figures show the raw lidar points (small black points), the lidar detections (dark green circles), the radar detections (red circles) and the fused sensor based track estimates (blue squares). In the figures it can be seen that while the radar and lidar occasionally produce distinct detections, the fused track estimates usually combine the detection qualities of both sensors. For example in Figures 3 (a), (b) and (c) the estimate within the dashed circle **A** is produced only by radar detections, the estimate in circle **B** only by lidar detections, and the estimate in circle **C** by both sensors. Note also that tracks are maintained during the absence of detections from each sensor, as can be seen in the case of the estimate within the dashed circle **A**.

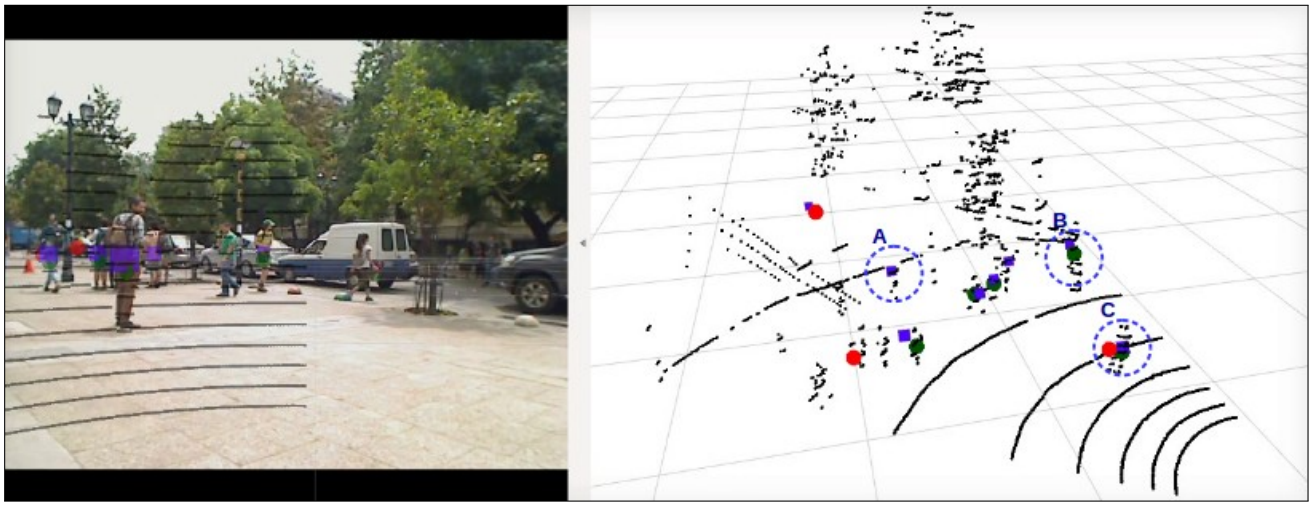
Figure 4 shows the results of the multi-sensor version of the δ-GMBP, which fuses the radar and lidar data, which is compared with its single-sensor equivalent using the OSPA metric [8]. The results based only on the radar had a very low performance, while those based only on the lidar show improved performance. The performance based on both sensors yielded the best results showing a slight improvement over the results based only on the lidar. This is because of the higher spatial noise and detection errors committed by the radar. A comparison between the δ-GMBP, m-MeMBer, and PHD filters, based on both the radar and the lidar sensors is shown in Figure 5. It can be seen that the δ-GMBP and m-MeMBer filters perform similarly. The failure of the PHD filter when processing both sensors can be explained by its documented forgetfulness when compared with the multi-Bernoulli filter [16].

V. Conclusions

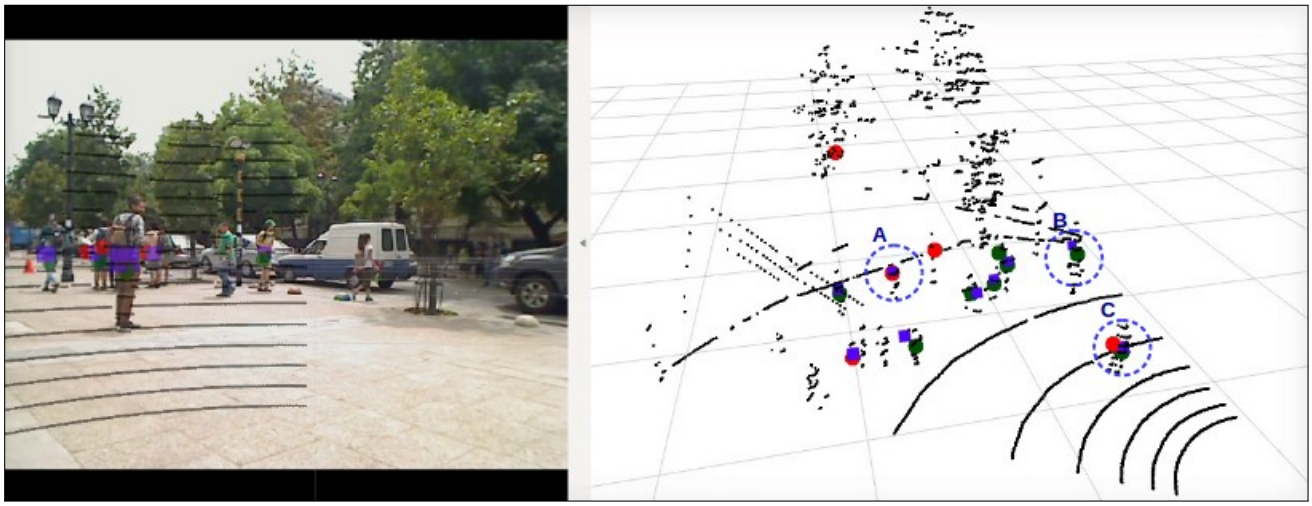
As expected, filtering using the multi-Bernoulli, δ-GMBP and m-MeMBer filters, yielded better performances than the PHD filter. A slight improvement of the δ-GMBP filter over



(a)



(b)



(c)

Figure 3. Images from radar and lidar tracking using the δ-GMBP filter. The left images show the projection of the detections and estimates on images recorded with a USB webcam. The right images show the 3D points, detections and estimates. Black points represent Velodyne range points, dark green circles represent people detection with the Velodyne, red circles radar detections, blue squares the estimates. Dashed blue circles labeled **A**, **B** and **C** indicate estimates from radar detections, lidar detections and both sensor detections.

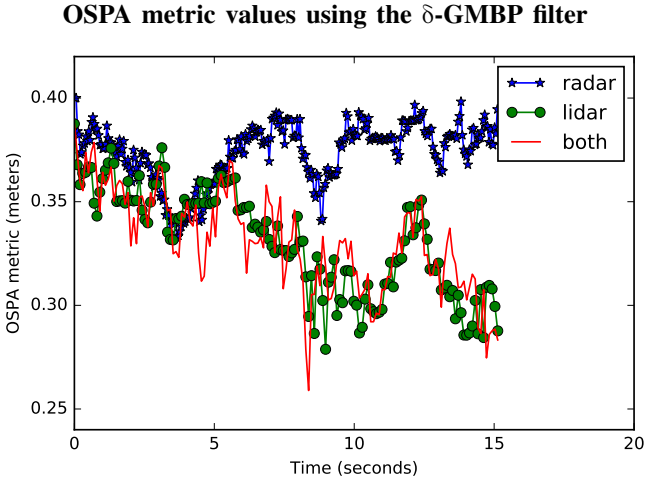


Figure 4. OSPA metric, with c=0.4 and p=2 using the δ-GMBP [13] filter, for tracking using the radar alone, the lidar alone and fusing both sensors.

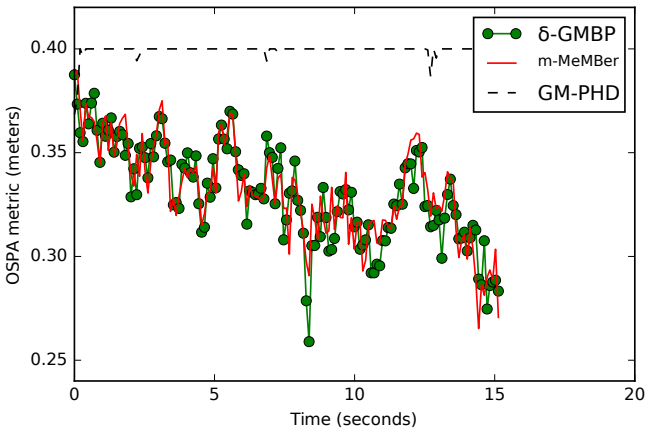


Figure 5. Comparison of δ-GMBP, m-MeMBer and PHD filters using the radar and the lidar asynchronously. δ-GMBP in green with circles, m-MeMBer in red straight line, PHD in black dashed line.

its m-MeMBer counterpart is also apparent in figure 5. The tracking performances of the δ-GMBP and m-MeMBer filters, based on both the radar and lidar data showed only slight improvements over the case when only lidar data was used. This is because the information provided by the radar contains much higher detection and spatial errors than that of the lidar. It should also be noted that the detection algorithm within the radar is not specifically designed to detect people. This is why the detections, which actually correspond to people, are sporadic in nature.

In order to improve filter performance, many improvements can be made, such as:

- Improving the state transition model, with a non-constant velocity model that takes into account the sudden stopping and change of direction of people.
- Modifying the likelihood model to include target descriptors such as the dimension of the targets, histograms, image descriptors or radar cross section.
- Augmenting the target state with the above descriptors. Thus, the probability of maintaining a correct track could be increased and wrong associations should be reduced.
- Using models for the probability of detection and for the background, taking into account the occlusions and the field of view of the sensors, in order to produce more realistic updates.
- Using the Gibbs sampler, for solving the cost assignment, as used in [7]. The Gibbs sampler has a computational complexity which is linear in the number of measurements, in contrast to Murty's algorithm which is cubic at best. This would allow the proposed algorithm to run in real time.

REFERENCES

[1] B.-T. Vo and B.-N. Vo, "Labeled Random Finite Sets and Multi-Object Conjugate Priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

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

[3] P. Harremoës, "Binomial and poisson distributions as maximum entropy distributions," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 2039–2041, 2001.

[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] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[6] G. Casella and E. I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[7] 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.

[8] 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, Aug. 2008.

[9] 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.

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

[11] K. G. Murty, "An Algorithm for Ranking all the Assignments in Order of Increasing Cost," *Operations Research*, vol. 16, no. 3, pp. 682–687, 1968.

[12] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of pearl's" belief propagation" algorithm," *IEEE Journal on selected areas in communications*, vol. 16, no. 2, pp. 140–152, 1998.

[13] 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.

[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

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

[16] K. Y. Leung, F. Inostroza, and M. Adams, "Multifeature-based importance weighting for the PHD SLAM filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2697–2714, 2016.