Bayesian Tracking of Multiple Objects with Vision and Radar

Michael Hoy, Chaoqun Weng, Junsong Yuan, and Justin Dauwels
School of Electrical and Electronic Engineering and ST Engineering – NTU Corporate Lab
Nanyang Technological University
Singapore

***Abstract*—This paper is concerned with a system for detecting and tracking multiple 3D bounding boxes based on information from multiple sensors. Our framework is built around an inference engine similar to the probability hypothesis density (PHD) filter, where the state space consists of stochastic bounding boxes with constant velocity dynamics. We outline measurement equations for two modalities (vision and radar). The result is a flexible inference system suitable for use on autonomous vehicles.**

I. Introduction

Recently, there has been significant interest in perception systems for autonomous vehicles (several reviews are available, see e.g. [1]–[3]). This paper is concerned with fusing detections from various sensor modalities into a list of 3D bounding boxes corresponding to objects surrounding the vehicle. Specifically, we consider radar and vision in this paper, though we are interested in general frameworks which can be adapted to different configurations.

In the following list we categorise some of the approaches available in the literature that address the object detection and tracking problem:

- *Specific processing pipelines:* There are several approaches based on pipelines for combining different sensors; these achieve good performance (see e.g. [4]–[6]). However we are interested in integrated probabilistic models that, for example, allow sensor configurations to be altered arbitrarily.
- *Optimisation based 3D bounding boxes:* Several papers compute bounding boxes as part of a probabilistic optimisation (see e.g. [7], [8]). While possibly not quite as temporally integrated as Bayesian filter based approaches, these papers contain many ideas that can be used in other contexts.
- *Bayesian filter based approaches:* This category ranges from approaches which track object centres (see e.g. [9], [10]), to approaches which add terms to maintain estimates of bounding boxes (see e.g. [11]–[13]), all the way to approaches which consider spatial extent as part of the core probabilistic update (see e.g. [14]–[17]). The latter seems to be a favourable way to address the problem.

The research was partially supported by the ST Engineering NTU Corporate Lab through the NRF corporate lab@university scheme.

Corresponding author: Michael Hoy (email: mch.hoy@gmail.com)

In this paper we consider a Bayesian filter approach with similarities to [7]–[9], [12], [14], [16].

In the multitarget tracking literature, there are two main approaches to computationally efficient multi-target tracking with Bayesian filters:

- *Data association methods:* this includes the joint probablistic data association filter (JPDAF), the multi hypothesis tracker (MHT), or more generally fixed measurement association schemes.
- *Random finite set methods:* the most common instantiation of this idea is some variant of the probability hypothesis density (PHD) filter, though other methods are available.

Reviews of these methods are available (see e.g. [3]). We implement an inference engine similar to the marginalized particle PHD filter. Marginalized particle filters have been used previously for object tracking problems (see e.g. [14]). This approach splits the state vector into linear and nonlinear components - in our case the linear components are the speed, bounding box dimensions and 2D position offset. While [14] employs a data association based design, marginalized particle PHD filters are available in the literature (see e.g. [18]). Approximate approaches are common for the multi-sensor PHD filter problem (see e.g. [19]); we will adopt the iterated-corrector.

For the visual detection measurement function, we will project the 3D bounding box onto the image frame (see e.g. [7], [8], [20]). In these approaches, a detector module analyses the camera feed and returns a list of detected 2D bounding boxes. The projected bounding box can then be linked to the detected bounding boxes using a Kalman update. We outline the necessary linearised update equations. For radar measurement function we compute an expected radar detection position conditioned on the object state (this idea is inspired by e.g. [21]). In this paper we use a particle labelling scheme, which has similarities to k-means clustering and track continuity [22].

To the best of our knowledge, we are are the first to use visual and radar information in a 3D bounding box PHD filter.

II. Details of Proposed Approach

A. General Formulation

We use an approach based on the multi-sensor probablity hypothesis density (PHD) filter, which is implemented with the marginalized particle filter approximation. We will adopt the choice of state vector from [14], except we make object speed a linear variable.

The state vector is split into discrete, linear and non-linear components, i.e. $\boldsymbol{x} = [x_\kappa, \boldsymbol{x}_n, \boldsymbol{x}_l]$ where $\boldsymbol{x}_n = [x_\theta, x_x, x_y]$, x_κ is the object class, and $\boldsymbol{x}_l = [x_v, x_h, x_w, x_l, x_{cw}, x_{cl}, x_e]$:

- x_θ is the heading in the global frame.
- x_x and x_y are the position in the local frame (distance in front of the ego vehicle and distance to the left of the ego-vehicle respectively).
- x_v is the speed in the global frame.
- x_h, x_w and x_l are the height, width and length of the object respectively.
- x_{cw} and x_{cl} are 2D offsets for the bounding box in the object frame, relative to $[x_x, x_y]$.
- x_e is the ground elevation near the object (relative to the ego vehicle).

In the marginalised particle filter representation, each particle with index i is represented by the following tuple:

$$\mathfrak{P}_k^i := (x_k^{\kappa,i}, \boldsymbol{x}_k^{n,i}, w_k^i, l_k^i, \boldsymbol{m}_k^i, P_k^i). \tag{1}$$

Here k represents the time index, $x_k^{\kappa,i}$ is the category, $\boldsymbol{x}_k^{n,i}$ is the non-linear part; w_k^i is the particle weight, l_k^i is the object track label; $\boldsymbol{m}_k^i$ is the mean of the linear part and P_k^i is the covariance of the linear part.

We will also refer to object tracks with index ℓ, where the state is averaged over the particles with the associated label:

$$\mathfrak{T}_k^\ell := (\boldsymbol{x}_k^{\kappa,\ell}, \boldsymbol{x}_k^{n,\ell}, \boldsymbol{x}_k^{l,\ell}, w_k^\ell), \tag{2}$$

where $\boldsymbol{x}_k^{\kappa,\ell}$ represents a categorical distribution over classes.

We assume the following: The ego-vehicle always has knowledge of the heading in the global frame[1] c_θ; we have acess to the change in the ego-vehicle position between updates $[\Delta x_x, \Delta x_y]$; and we have knowledge of the ego-vehicle velocity $\bar{\boldsymbol{v}}$.

Each measurement is expressed as a measurement matrix H, an offset $\boldsymbol{h}$ and a covariance R. Note that H specifies the linearised relationship between the linear states and the measurements. This represents the linear state conditioned on the non-linear state, such that $H\boldsymbol{x}_l \sim N(\cdot; \boldsymbol{h}, R)$. We also introduce the following intermediate variables: $x_{\theta,v} := x_\theta - c_\theta$ is the object heading in ego-vehicle frame, and $x_\phi := \text{atan2}(x_y, x_x) + \pi - x_{\theta,v}$ is the heading to the ego-vehicle from the object in the object frame.

[1]For simplicity we express all headings in the global frame. Note that in practice the heading does not need to accurately represent the actual compass heading (it is sufficient that it is self-consistent over time).

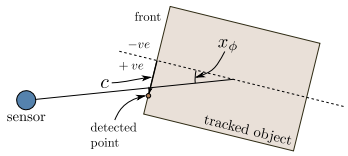


Fig. 1. The relationship between the angle to the sensor from the object in the object frame x_ϕ, and the displacement c of the detected point from the object centreline.

B. Radar Detections

An automotive radar sensor generally returns one 2D point corresponding to the position of each detected object[2]. There is some work which relates radar to bounding boxes (see e.g. [21]) and we make an assumption in a similar vein[3]:

Assumption 2.1: Within the range $\frac{-\pi}{4} < x_\phi < \frac{\pi}{4}$, the expected position of the radar detection is on the front face of the object. The displacement along the front face of the object is given as $c := \frac{2x_w x_\phi}{\pi}$ (relative to the forward centreline of the object; see Fig. 1).

Note the H matrix is dependent on the angle $\boldsymbol{x}_\phi$; the subscript indicates the relevant range. For example, the left side is given as:

$$H_{\frac{-3\pi}{4}:\frac{-1\pi}{4}} = \begin{bmatrix} 0 & 0 & 0 & b & 0 & 1 & 0 \\ 0 & 0 & -0.5 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{3a}$$

$$\text{where } b = \frac{\text{mod}\ (\boldsymbol{x}_\phi, \frac{-\pi}{4}, \frac{\pi}{4})}{\frac{\pi}{2}}. \tag{3b}$$

The offset $\boldsymbol{h}$ is given as:

$$\boldsymbol{h} = B_{-\boldsymbol{x}_{\theta,v}} \cdot \begin{bmatrix} x_m - \boldsymbol{x}_x \\ y_m - \boldsymbol{x}_y \end{bmatrix}, \tag{4}$$

where $[x_m,\ y_m]$ is detected point; $\text{mod}\ (x, a, b)$ applies the modulo operator to ensure $a \leq x < b$; B_θ is the 2D rotation matrix and the measurement covariance R is constant.

Note for the results presented here, radar velocity was not used as it occasionally gives incorrect measurements; further investigation is needed to characterise this.

C. Visual Detections

We will make the following assumptions:

- We have the position of detected bounding boxes in the image frame, together with uncertainties on each face and the vertical centerline[4], so that the mean and variance is expressed as $[\eta_d, \rho_d]$, where $d \in \{up, down, left, right\}$. For simplicity, we assume the uncertainties are independent[5].

[2]This seems to hold for small objects like pedestrians, larger objects with surfaces oriented perpenduclarity to the sensor may lead to multiple points.

[3]In future work we plan to investigate the implications of this assumption in more detail.

[4]In our experience the position of the sides of visually detected bounding boxes do not always accuarately correspond to the 3D bounding box, so we assign more weight to the verticle centerline than the sides.

[5]In future work, the uncertainty could be estimated adaptively for each detection. This would be helpful if for instance only part of a large object is detected occasionally.

- We have access to a transformation from the local frame to a cordinate frame centered on the camera T_{cw}. For simplicity we assume it is fixed.
- We have access to the camera matrix C (i.e. a matrix that maps from the homogeneous coordinates onto pixel coordinates).

If we consider an arbitrary point $\boldsymbol{p}_{ref}$, we can find the projected position $\tilde{\boldsymbol{f}}_{\boldsymbol{p}_{ref}} := [\tilde{f}_u, \tilde{f}_v, f_d]$. The actual pixel coordinates $\boldsymbol{f}_{ref} := [f_u, f_v]$ are given by $f_u := \frac{\tilde{f}_u}{f_d}$ and $f_v := \frac{\tilde{f}_v}{f_d}$. We also find the Jacobian $J_{\boldsymbol{p}_{ref}}$:

$$\tilde{\boldsymbol{f}}_{\boldsymbol{p}_{ref}} = CT_{cw}\boldsymbol{p}_{ref}, \tag{5a}$$

$$J_{\boldsymbol{p}_{ref}} := \frac{1}{f_d}\begin{bmatrix} 1 & 0 & -f_u \\ 0 & 1 & -f_v \end{bmatrix} CT_{cw}. \tag{5b}$$

The general steps are as follows:

- Consider the eight corners of the *mean* bounding box (with position vector $\boldsymbol{p_i}$), and project each onto the camera frame (with pixel coordinate vector $\boldsymbol{f_i}$):

$$\boldsymbol{p_i} = [x_x, x_y, x_e] + L_{\boldsymbol{i}}\boldsymbol{x}_l, \tag{6a}$$

$$\boldsymbol{i} \in \{left, right\} \times \{front, back\} \times \{up, down\}, \tag{6b}$$

  where $L_{\boldsymbol{i}}$ relates $\boldsymbol{x}_l$ to the local frame around the vehicle (definition in (9)).
- Find the most extreme side in each direction in the camera frame, i.e. for each $d \in \{up, down,\ left, right\}$:

$$\boldsymbol{i}_d = \begin{cases} \arg\max_{\boldsymbol{i}} A_d \boldsymbol{f_i} & d \in \{down, right\} \\ \arg\min_{\boldsymbol{i}} A_d \boldsymbol{f_i} & d \in \{up, left\} \end{cases}, \tag{7a}$$

$$\text{where } A_d = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & d \in \{left, right\} \\ & d \in \{up, down\} \end{cases}. \tag{7b}$$

  Let $\boldsymbol{p}_d := \boldsymbol{p}_{\boldsymbol{i}_d}$, $L_d := L_{\boldsymbol{i}_d}$, etc.
- For each side in the camera frame, find the linearised matrices:

$$H_d = A_d J_{\boldsymbol{p}_d} L_d, \tag{8a}$$

$$h_d = \eta_d - A_d(\boldsymbol{f}_d - J_{\boldsymbol{p}_d} L_d \boldsymbol{x}_l), \tag{8b}$$

$$R_d = \rho_d. \tag{8c}$$

- Repeat the previous step to find the update equation for the vertical centerline.

D. Transition Equations

We assume all vehicles follow constant velocity modified unicycle dynamics, which is a reasonable expectation for vehicles manoeuvring at low speed.

$$\boldsymbol{x}^n_{k|k-1} = \boldsymbol{f}_{\boldsymbol{x}_n} + \boldsymbol{e}_n, \quad \boldsymbol{e}_n \sim N(\cdot; 0, Q_n), \tag{10a}$$

$$\boldsymbol{f}_{\boldsymbol{x}_n} = \begin{bmatrix} x^\theta_{k-1} \\ x^x_{k-1} + \Delta x_x \\ x^y_{k-1} + \Delta x_y \end{bmatrix}, \tag{10b}$$

$$\boldsymbol{x}^l_{k|k-1} = F_{\boldsymbol{x}_l}\boldsymbol{x}^l_{k-1} + \boldsymbol{e}_l, \quad \boldsymbol{e}_l \sim N(\cdot; 0, Q_l), \tag{10c}$$

where $[\Delta x_x, \Delta x_y]$ is the change in the position of the tracked object due to the change in the ego-vehicles position, and Δt is the time step. $F_{\boldsymbol{x}_l}$ is the identity matrix except for one Δt term linking x_v and x_{cl}.

E. Probabilistic Update

For RFS filters it is typically assumed that the measurement function normalisation constant (NC) is known. Unlike the single target Bayesian filter, where the NC cancels out, in RFS filters it is needed for a clutter correction. One issue is that due to model mismatch, we may only want to perform inference with $g(\cdot)$ up to an unknown scale factor. There is some research into PHD filters with unknown clutter rates (see e.g. [19], [23]). Here objects known as "clutter generators" are modelled as a special type of detected object. We take inspiration from from this idea, and assume the clutter has the same dynamic properties as the detected objects. This allows us to consider an alternative equation of the following form (using notation from [18]):

$$v_k(x) = (1 - p_d(x))v_{k|k-1}(x) + \sum_{z \in Z_k} \frac{p_d(x)g(z|x)v_{k|k-1}(x)}{(\kappa_p(z) + 1) \cdot \int p_d(\xi)g(z|\xi)v_{k|k-1}(\xi)d\xi}, \tag{11}$$

where $\kappa_p(z)$ is a "proportional clutter" rate. Note that we also employ separate object classes for clutter; these may help to model situations where clutter is not temporally independent.

We assume each sensor has a constant proportional clutter rate κ_p (as described in Sec. II-E), and that the detection probability $p_d(\boldsymbol{x}_n)$ is constant for each sensor, given the nominal position $[x_x, x_y]$ is inside the sensor's field of view (otherwise $p_d(\boldsymbol{x}_n)$ is zero).

F. Measurement Association

Measurement association is performed for the purpose of initiating new track labels (and thus providing initial track estimates for k-means based state segmentation; see Sec. II-G).

The following steps will be used to perform measurement association on each sensor modality individually:

- Compute a dissimilarity function $S(\mathfrak{D}^j_k, \mathfrak{T}^\ell_k)$ that relates each detection $\mathfrak{D}^j_k$ to each object track $\mathfrak{T}^\ell_k$. If $S(\mathfrak{D}^j_k, \mathfrak{T}^\ell_k) < \epsilon_1$, the object track $\mathfrak{T}^\ell_k$ and detection $\mathfrak{D}^j_k$ can be associated.
- For each unassociated detection, we create a new object track label ℓ_{new} with an initial state estimate $\mathfrak{T}^{\ell_{new}}_k$ estimated from the detection information $\mathfrak{D}^j_k$.

For the function $S(\cdot, \cdot)$, we use horizontal distance for radar detections and pixel displacement for camera detections.

G. State Segmentation and Particle Labelling

We employ the following steps partly based on k-means clustering (inspired by [22]):

- For each particle, if the distance to the current label is below a threshold ϵ_3, retain the current label. Else find the most likely object track label:

$$a_{\boldsymbol{i}} = \begin{cases} +1 & left \in \boldsymbol{i} \\ -1 & right \in \boldsymbol{i} \end{cases}, \quad b_{\boldsymbol{i}} = \begin{cases} +1 & front \in \boldsymbol{i} \\ -1 & back \in \boldsymbol{i} \end{cases}, \quad c_{\boldsymbol{i}} = \begin{cases} +1 & up \in \boldsymbol{i} \\ 0 & down \in \boldsymbol{i} \end{cases}, \tag{9a}$$

$$L_{\boldsymbol{i}} = \begin{bmatrix} 0 & 0 & b_{\boldsymbol{i}} \sin(\boldsymbol{x}_\theta) & a_{\boldsymbol{i}} \cos(\boldsymbol{x}_\theta) & \sin(\boldsymbol{x}_\theta) & \cos(\boldsymbol{x}_\theta) & 0 \\ 0 & 0 & b_{\boldsymbol{i}} \cos(\boldsymbol{x}_\theta) & -a_{\boldsymbol{i}} \sin(\boldsymbol{x}_\theta) & \cos(\boldsymbol{x}_\theta) & -\sin(\boldsymbol{x}_\theta) & 0 \\ 0 & c_{\boldsymbol{i}} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9b}$$

$$\ell_k^i = \arg\min_{\ell} D(\boldsymbol{x}_k^i, \boldsymbol{x}_k^\ell). \tag{12}$$

Here $D(\cdot,\cdot)$ represents a distance metric between two state space samples. If the distance is below a threshold ϵ_4, we can adopt the best object track. Otherwise the particle is unlabelled.

- Recompute object track state estimate from updated labels.
- Perform track maintenance:
  - Remove track labels whenever $w_k^\ell = 0$.
  - Merge object tracks whenever $D(\boldsymbol{x}_k^{\ell_1}, \boldsymbol{x}_k^{\ell_2}) < \epsilon_2, \forall \ell_1, \ell_2 \in \ell$.
- Optionally repeat until convergence (this was not needed for the experiments presented here).

H. Processing Steps

The algorithm consists of the steps shown in Fig. 2 (note the marginalised particle filter updates are generally based on [18], though the equations are simplified by both the absence of linear dynamics and the use of the proportional clutter as mentioned in Sec. II-E):

III. Practical Aspects

For the results presented in this paper we restrict ourselves to pedestrian and cyclist detection. We employ the faster RCNN object detector [24] [6].

For both classes (pedestrian and vehicle) we add "clutter versions" of the classes. These are the same as the non clutter versions, except they have a different birth weight, are assumed to be only detectable by radar, and are assumed to be stationary (thus have a different Q_l). They are included in the object tracks, which allows each object to be assigned a given probability of being clutter.

We only store one off-diagonal element (between x_v and x_{cl}) of the covariance matrix to increase processing speed. This did not seem to significantly affect the results, through further work could better quantify this.

We only add new birth particles in the vicinity of detections, since particles far away from detections are quickly discarded. As an approximation, we add n_b particles each with a weight of w_b around each detection. So long as w_b is small, this should have minimal bias on the results. Note that w_b is different for clutter and object classes.

[6]This was pretrained on VOC 2007 dataset [25] and fine-tune the model using Caltech pedestrian detection dataset [26]. In the experiments, the object detector threshold is set to 0.5 and the intersection-over-union threshold in the non-maxima-suppression step is set to 0.5. After the object detection, the results are further smoothed by tracklet algorithm proposed in [27].

κ_p (camera)	1.5
p_d (camera)	0.25
ϵ_1 (camera)	10 pixels
ρ_d (camera; centre, bottom)	10
ρ_d (camera; sides, top)	1000
κ_p (radar)	1000
p_d (radar)	0.15
ϵ_1 (radar)	$1m$
R (radar: position)	**diag**($\begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$)
n_p	1000
n_b	100
w_b (object)	10^{-8}
w_b (clutter)	10^{-6}
ϵ_2	1
ϵ_3	8
ϵ_4	1
Q_l (pedestrian)	**diag**($\begin{bmatrix} 0.5, 0, 0, 0, 10^{-3}, 10^{-3}, 0 \end{bmatrix}$)
Q_l (pedestrian, clutter)	**diag**($\begin{bmatrix} 0, 0, 0, 0, 10^{-1}, 10^{-1}, 0 \end{bmatrix}$)
$\boldsymbol{m}_p$ (pedestrian)	$[0, 1.5, 0.5, 0.5, 0, 0, 0]^\intercal$
P_p (pedestrian)	**diag**($\begin{bmatrix} 0.5, 10^{-6}, 10^{-6}, 10^{-6}, 1, 1 \end{bmatrix}$)
Q_l (cyclist)	**diag**($\begin{bmatrix} 50, 0, 0, 0, 10^{-3}, 10^{-3}, 0 \end{bmatrix}$)
Q_l (cyclist, clutter)	**diag**($\begin{bmatrix} 0, 0, 0, 0, 10^{-1}, 10^{-1}, 0 \end{bmatrix}$)
$\boldsymbol{m}_p$ (cyclist)	$[0, 1.8, 0.9, 1.6, 0, 0, 0]^\intercal$
P_p (cyclist)	**diag**($\begin{bmatrix} 400, 10^{-6}, 10^{-6}, 10^{-6}, 1, 1, 0 \end{bmatrix}$)
p_s	0.99

TABLE I
PARAMETERS USED FOR EXPERIMENTS.

The inference was implemented in C++ using OpenCV. Without optimisation, computation takes ≈ 400 ms per update.

We employed a hysteresis threshold to select objects based on weight and clutter probabilities. Finding stable estimates of the heading of objects presented some difficulties, and we use additional smoothing to find the final estimated heading.

IV. Experiments

A. Overview

Since we could not locate suitable public datasets that include radar data, we performed experiments on a proprietry dataset. The camera used was the JAI AD-080 with a resolution of 1024 x 768. Calibration was performed using a checkerboard pattern and the OpenCV calibration suite. The radar was the Delphi ESR system.

B. Qualitative Results

Sample results are shown in Fig. 3. In each figure, the red box shows the result of the image object detector, and the green box shows the detected objects in the bounding box filter. It can be seen that most objects are correctly detected, at least in cases where the visual bounding boxes where correctly identified. The track management was able to assign static ID's to objects in most situations.

1) Associate the detections and create new object track labels (as specified in Sec. II-F).
2) Apply the transition equations to each particle:
   a) Update particle weights:

$$w^i_{k|k-1} = p_s w^i_{k-1}. \tag{13}$$

   b) Compute $\boldsymbol{f}_{\boldsymbol{x}_n}$ and $F_{\boldsymbol{x}_n}$ based on $\boldsymbol{x}^{n,i}_{k-1}$ according to Sec. II-D.
   c) Sample $\boldsymbol{x}^{n,i}_{k|k-1}$:

$$\boldsymbol{x}^{n,i}_{k|k-1} \sim N(\cdot; \boldsymbol{f}_{\boldsymbol{x}_n}, Q_n + F_{\boldsymbol{x}_n} P^i_{k-1} F^{\intercal}_{\boldsymbol{x}_n}). \tag{14}$$

   Here $N(\boldsymbol{x}; \boldsymbol{m}, P)$ is the standard Gaussian density.
   d) Predict $\boldsymbol{m}^i_{k|k-1}$ and $P^i_{k|k-1}$:

$$\boldsymbol{m}^i_{k|k-1} = F_{\boldsymbol{x}_l} \boldsymbol{m}^i_{k-1}, \tag{15a}$$
$$P^i_{k|k-1} = F_{\boldsymbol{x}_l} P^i_{k-1} F^{\intercal}_{\boldsymbol{x}_l}. \tag{15b}$$

3) Apply the following steps for each sensor modality sequentially:
   a) For each particle i and each detection group j (which comprises the set of update equations $\mathfrak{H}_j \ni (H, \boldsymbol{h}, R)$, compute the Kalman update:

$$w''_{ij} = w^i_{o|k-1} \prod_{(H,\boldsymbol{h},R)\in\mathfrak{H}_j} N(h; H\boldsymbol{m}^i_{o|k-1}, S_{H,R}), \tag{16a}$$

$$\boldsymbol{m}'_{ij} = \boldsymbol{m}^i_{o|k-1} + \sum_{(H,\boldsymbol{h},R)\in\mathfrak{H}_j} K_{H,R} \cdot (\boldsymbol{h} - H\boldsymbol{m}^i_{o|k-1}), \tag{16b}$$

$$P'_{ij} = P^i_{o|k-1} - \sum_{(H,\boldsymbol{h},R)\in\mathfrak{H}_j} K_{H,R} S_{H,R} K^{\intercal}_{H,R}, \tag{16c}$$

$$S_{H,R} = H P^i_{o|k-1} H^{\intercal} + R, \tag{16d}$$

$$K_{H,R} = P^i_{o|k-1} H^{\intercal} S^{-1}_{H,R}. \tag{16e}$$

   b) Normalise the weight contribution for each particle:

$$w'_{ij} = \frac{w''_{ij}}{(\kappa_p + 1) \cdot \sum_{i'} w''_{i'j}}. \tag{17}$$

   c) Sample the linear update for each particle:

$$(\boldsymbol{m}^i_k, P^i_k) = \begin{cases} (\boldsymbol{m}^i_{o|k-1}, P^i_{o|k-1}) \\ \text{prob} \propto w^i_{o|k-1}(1 - p_d) \\ \\ (\boldsymbol{m}'_{ij}, P'_{ij}) \\ \text{prob} \propto w'_{ij} \quad \forall j \end{cases}. \tag{18}$$

   d) Compute the final particle weight:

$$w^i_k = w^i_{o|k-1} \cdot \left(1 - p_d(\boldsymbol{x}^{n,i}_{o|k-1})\right) + \sum_j w'_{ij}. \tag{19}$$

   e) Update object track labels for each particle and compute EAP estimates for each object track (see Sec. II-G).
4) Resample to a constant number of particles n_p[a].

[a]We use an adaptive resampling scheme, where quotas are given to different objects and classes, and the probability of selection is proportional to a power function of the weight.

Fig. 2. Algorithm overview.

C. Failure Cases

In this section we outline some examples of scenarios where the system did not work as expected.

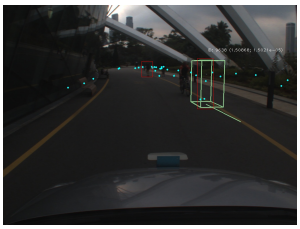
Fig. 4 shows the detection of two pedestrians with a single bounding box, despite two image frame detections being presented by the image object detection module. This was caused by an incorrect decision by the track management system. This could be resolved by improving the logic to generate new object tracks. In particular, the track merging should recognise when there are in fact two objects, perhaps by considering the PHD weight. It also shows detection of a non-pedestrian object. This was caused by a false detection in the visual object detector. There are also cases where the visual object detector misses a pedestrian. These cannot be resolved through improvements to the tracking algorithm, and could only be resolved via improvements to the bottom up visual object detector.

V. Future Work

The following areas are possible avenues for investigation in the future:

- *Other sensors:* we plan to integrate both lidar detections and optic flow information.
- *Improvements to track management:* many of the failure cases seem to be caused by bad track management. We will investigate more sophisticated, possibly model based approaches, which may give better performance. Ideally good track management could be achieved with few heuristic rules.
- *Radar system identification:* it may be possible to perform experiments to characterise how automotive radar is reflected off various types of objects.
- *Elevation as a state variable:* This could be useful in cases where the ground plane is not approximately flat. However it is not clear whether such a system is observ-

(a) (b)



(c)

Fig. 3. Detection of multiple pedestrians and cyclist.

(a) (b)

Fig. 4. (a) Detection of two pedestrians with a single bounding box. (b) Detection of a non-pedestrian object.

able with just visual and radar information.

VI. Conclusion

We propose a PHD filter framework for tracking 3D bounding boxes using multi-sensor information. Measurement equations for each modality (i.e. radar and visual object detections) are provided. We also proposed a "proportional clutter" PHD filter for measurement functions with unknown normalisation constants. Experimental results show that the system is viable. While there are several cases where the system did not operate as expected, we plan to address these in future work.

References

[1] S. Sivaraman and Mohan M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *Intelligent Transportation Systems, IEEE Transactions on*, 14(4):1773–1795, 2013.

[2] D. Geronimo, A. M Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1239–1258, 2010.

[3] C. Stiller, F. P. Leòn, and M. Kruse. Information fusion for automotive applications–an overview. *Information fusion*, 12(4):244–252, 2011.

[4] C. Premebida and Urbano J. C. Nunes. Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection. *The International Journal of Robotics Research*, page 0278364912470012, 2013.

[5] L. Oliveira, U. Nunes, P. Peixoto, M. Silva, and F. Moita. Semantic fusion of laser and vision in pedestrian detection. *Pattern Recognition*, 43(10):3648–3659, 2010.

[6] K. Kidono, T. Naito, and J. Miura. Reliable pedestrian recognition combining high-definition lidar and vision data. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1783–1788. IEEE, 2012.

[7] S. Song and M. Chandraker. Joint sfm and detection cues for monocular 3d localization in road scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3734–3742, 2015.

[8] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):882–897, 2013.

[9] D. Meissner, S. Reuter, E. Strigel, and K. Dietmayer. Intersection-based road user tracking using a classifying multiple-model phd filter. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):21–33, 2014.

[10] L. Spinello, R. Triebel, and R. Siegwart. Multiclass multimodal detection and tracking in urban environments. *The International Journal of Robotics Research*, 29(12):1498–1515, 2010.

[11] K. Schueler, T. Weiherer, E. Bouzouraa, and U. Hofmann. 360 degree multi sensor fusion for static and dynamic obstacles. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 692–697. IEEE, 2012.

[12] Y. Yeo, X. Zhang, and R. Yang. A perception system for obstacle detection and tracking in rural, unstructured environment. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE, 2014.

[13] S. Sivaraman and M. M. Trivedi. Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1249–1254. IEEE, 2011.

[14] A. Petrovskaya and S. Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139, 2009.

[15] A. Vatavu, R. Danescu, and S. Nedevschi. Stereovision-based multiple object tracking in traffic scenarios using free-form obstacle delimiters and particle filters. *Intelligent Transportation Systems, IEEE Transactions on*, 16(1):498–511, 2015.

[16] K. Granstrom, S. Reuter, D. Meissner, and A. Scheel. A multiple model phd approach to tracking of cars under an assumed rectangular shape. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE, 2014.

[17] C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a phd filter. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[18] Y. Petetin, M. Morelande, and F. Desbouvries. Marginalized particle phd filters for multiple object bayesian filtering. *IEEE Trans. Aerosp. Electron. Syst*, 50(2):1182–1196, 2014.

[19] R. Mahler. statistics 102?for multisource-multitarget detection and tracking. *Selected Topics in Signal Processing, IEEE Journal of*, 7(3):376–389, 2013.

[20] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

[21] H. Cho, Y. Seo, V. Kumar, and R. R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1836–1843. IEEE, 2014.

[22] D. E. Clark and J. Bell. Multi-target state estimation and track continuity for the particle phd filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(4):1441–1453, 2007.

[23] R. Mahler, B. Vo, and B. Vo. Cphd filtering with unknown clutter rate and detection profile. *Signal Processing, IEEE Transactions on*, 59(8):3497–3513, 2011.

[24] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[25] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[26] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[27] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.