Title

Siddharth Ratapani Navin, Maani Ghaffari

***Abstract*— In this work, we present a highly scalable probabilistic method for vehicle detection and tracking, based on computer vision and multi-object tracking algorithms. The developed pipeline implements mono-camera based motion estimation through scene flow, vehicle detection through a pre-trained Regional Convolutional Neural Network (RCNN) and tracking through a *Probability Hypothesis Density* (PHD) Filter [13]. Most current methods on tracking multiple objects incorporate propagating sensor data-association hypotheses in time, which in turn make them dependent on correct data association in the observations. This lead to poor performance in tracking, where the algorithm diverges and produces erroneous results. In this work, we propose the use of the PHD Filter that implements a recursive algorithm for jointly estimating the time varying number of targets in the presence of data association uncertainty, detection uncertainty, noise and false alarms. A key challenge to this approach is efficiently localizing objects in the crowded scenes and under occlusions. Instead of solely relying on 2D object detection algorithms, our method leverages both mature 2D object detectors such as *Mask R-CNN* [6] and visual odometry ques through *Gunnar Farneback's Optical Flow* [3]. We are able to achieve good tracking results in most cases, including crowded and/or occluded environments. The pipeline has been evaluated on the *CityScape* [2] dataset.**

I. Introduction

The objective of Multi-Object tracking is to jointly estimate, at each time step, the number of targets and their states from a sequence of noisy and cluttered observation sets. In Multi-Object tracking, we observe variation in the states of the targets with time, however, the number of targets also changes due to random appearance and disappearance. We also observe scenarios where the detector is incapable of detecting all targets. An intrinsic problem in the field of tracking objects is observation data association. No deterministic method exists yet to associate each sensor reading with the target, hence we must resolve to combinatorial methods. Due to this nature, the data association problem makes up the bulk of the computational load in multi-target tracking algorithms. We use an analytic solution to the PHD recursion for linear Gaussian target dynamics and Gaussian birth model which is analogous to the Kalman filter as a solution to the single-target Bayes filter. Moreover, closed form recursions for the weights, means, and co-variances of the constituent Gaussian components are derived. The resulting filter propagates the Gaussian mixture posterior intensity in time as measurements arrive. Instead of tracking low level features like most conventional tracking algorithms that provide very little semantic information about the scene, we implement deep learning based object detection and segmentation to track the semantics such as object masks, class information and bounding boxes.

S. Ratapani Navin and M. Ghaffari are with the College of Engineering, University of Michigan, Ann Arbor, MI 48109 USA {sidnav, maanigj}@umich.edu.

A. Contributions

B. Outline

II. Related Work

Multiple Hypotheses Tracking (MHT) and its variations concern the propagation of associating hypotheses in the time domain and weighing these observations by their association probabilities. The joint probabilistic data association filter (JPDAF) [1], [4], the probabilistic MHT (PMHT) [12], and the multi-target particle filter [7] use observations weighted by their association probabilities. Alternative formulations that avoid explicit associations between measurements and targets include Symmetric Measurement Equations [8] and Random Finite Sets (RFS) [9], [5].

In the RFS formulation, the collection of individual targets is treated as a set-valued state, and the collection of individual observations is treated as a set-valued observation. Novel RFS-based filters such as the PHD filter and their implementations have generated substantial interest. The PHD filter propagates the first-order statistical moment of the RFS of states in time. This approximation was developed to alleviate the computational intractability in the multi-target Bayes filter, which stems from the combinatorial nature of the multi-target densities and the multiple integrations on the (infinite dimensional) multi-target state space.

Other work on multi-hypothesis tracking tend to fail when the target object changes poses, because such changes often lead to mismatch between the appearance model and the learned one. These approaches primarily rely on tracking low level features which provide low semantics. Bo Zhang et al.[14] discuss an ensemble color feature model which takes the physiological structure of the object into consideration, and takes advantage of color histogram information under several color spaces, including the RGB, normRGB, HSV, and Lab.

III. Methodology

A. Mask R-CNN based Instance Segmentation

Regional Convolutional Neural networks (R-CNNs) solve the problem of detection and localization of the object in the image through bounding boxes and respective classification of objects in the image. Mask R-CNN [6] is the newest member in the family of RCNN's which takes a step further by implementing a pixel-wise segmentation of the objects in the image. It does this by adding a branch to Faster R-CNN [11] that outputs a binary mask that says whether or

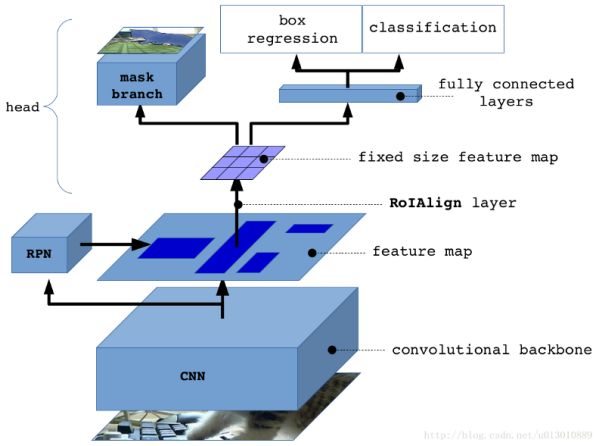

Fig. 1. Mask R-CNN architecture : a fully connected network added to Faster R-CNN [11]

not a given pixel is part of an object *(refer Fig.1)*. It adopts the same two-stage procedure,which consists of a Region Proposal Network [11] followed by the Regressor, Classifier and Fully Convolutional Network.

We modified the Mask R-CNN network to classify just 2 classes : (i) vehicle or (ii) background. The frames from the video feed serve as the input to the network. For each frame we obtain the following:

- **Masks** - This is a 3 dimensional array, each 2 dimensional instance of this array corresponds to the masked image of the car it is associated with.
- **Bounding boxes** - These are minimum area rectangles that are built around each vehicle.

We use image sequences, for testing the Modified Mask R-CNN and save the output results to a file, for ease of use and computational efficiency. Since we do not have access to a powerful GPU, we use this file to test the performance of the implementation of the PHD Filter. The bounding box output serves as the sensor measurement or the *Sensor Model* in our tracking problem and the formulation will be outlined in a later section.

B. Dense Optical Flow

Scene flow (or *Optical Flow*) is the pattern of apparent motion of objects or other features in a visual scene caused by the relative motion between an observer and a scene. Optical flow is based of the following assumption -

$$I(\hat{x}, t) = I(\hat{x} + \hat{u}, t + 1) \tag{1}$$

where $I(x, t)$ is image intensity as a function of space $x = (x, y)^T$ and time t, and $u = (u_1, u_2)^T$ is the 2D velocity. To derive 2D velocity u , we consider the Taylor series approximation-

$$\nabla I(\hat{x}, t)\hat{u} + I_t(\hat{x}, t) = 0 \tag{2}$$

The above equation is the basis for the optical flow velocity-u estimation between frames. In our implementation, a *dense optical flow* computation is done based on Gunnar Farneback-âĂŹs algorithm [3]. We calculate the mean displacements for each region of interest using the mask output on the flow image and these act as the *Motion Model* for the PHD Filter, as explained in a later section.

Fig. 2. *from left to right, top to bottom* (i) the original frame of the image sequence at t = 0, (ii) the original frame of the sequence at t = 1, (iii) the result of the detection using Mask R-CNN on the first image, and (iv) the result of dense optical flow between the 2 frames

C. PHD Filter

1) Single Object Tracking with Kalman Filter: In many dynamic state estimation problems, the state is assumed to follow a Markov process [1] on the state space $X \in R^{nk}$, with transition density $f_{k|k-1}$, i.e. given a state x_{k-1} at time $k-1$, the probability density of a transition to the state x_k at time k is $f_{k|k-1}(x_k|x_{k-1})$. This mathematical model for uncertainty propagation is called the *Motion Model* in Probabilistic Robotic literature. We will discuss our implementation of this model later on.

This Markov process is partially observed in the observation space $Z \in R^z$, it is modelled by the likelihood function g_k, i.e. given a state x_k at time k, the probability density of observation $z_k \in Z$ is $g_k(z_k|x_k)$. This mathematical model is called as the likelihood model or *Sensor model* [1]. The probability density of the state x_k at time k given all observations $z_{1:k} = (z_1, ..., z_k)$ up to time k, is denoted by $p_k(x_k|z_{1:k})$ and is called the posterior density (or filtering density) at time k. From an initial density p_0, the posterior density at time k can be computed using the Bayes recursion

$$p_{k|k1}(x_k|z_{1:k-1}) = f_{k|k-1}(x_k|x_{k-1})p_{k-1}(x_{k-1}|z_{1:k-1})dx \tag{3}$$

$$p_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)p_{k|k-1}(x_k|z_{1:k1})}{\int g_k(z_k|x_k)p_{k|k-1}(x_k|z_{1:k-1})dx} \tag{4}$$

Estimates of the state at time k can be obtained using either the MMSE (Minimum Mean Squared Error) criterion or the MAP (Maximum A Posteriori) criterion [1].

2) Random Finite Set Formulation: Suppose at time t_k , where $k = 0, 1, 2...$, there are n_k objects (targets) with states $x_{k,1}, ..., x_{k,n_k}$, taking values in the state space $X \subset R^{nk}$. Both the number of targets n_k and their individual states in X are random and time varying. The multi-target state at k, represented by a finite set $X_k = x_{k,1}, ..., x_{k,n_k} \in F(X)$. $F(X)$ is a set of finite subsets of X.

The detection process is imperfect meaning that some of the targets in X_k are detected, while the others are missed. In addition, the detector typically creates false detections. Suppose measurement sets are available at time $k = 1, 2, ...$ A measurement set at time k contains m_k elements, each taking a value in the observation space $Z \subset R^z$. Then the multi-target observation set is represented by another set $Z_k = z_{k,1}, ..., z_{k,m_k} \in F(Z)$, where $F(Z)$ is a set of finite subsets of Z. Since an RFS is nothing but a finite-set random variable, the usual probabilistic descriptors of a random variable, such as the Probability Density Function and its statistical moments, can be defined for it.

The intensity function (also known as the *probability hypothesis density* or *PHD*) of an RFS X is defined as its first statistical moment:

$$D(x) = \mathbb{E}\{\delta_X(x)\} = \int \delta_X(x) f(X) \delta X \tag{5}$$

where $\delta_X(x)$ is the set Dirac delta function [1]. In simpler terms, the PHD is the value associated with the expected number of targets in the state. The PHD filter propagates this density. The set of observations carry no information about which target generated the observation, since there is no ordering on the respective collection of states and measurements at time k, they can be represented as a RFS.

In a single-target system, uncertainty is characterized by modelling the state x_k and measurement z_k as random vectors. Analogously, uncertainty in a multi-target system is characterized by modelling the multi-target state X_k and multi-target measurement Z_k as RFS.

3) Linear Gaussian Mixture Formulation: In the implementation of the PHD filter we assume that the each state of the target we are tracking is defined by an independent Gaussian distrubution with mean and variance μ_k, Σ_k,

$$p_k(x_k|z_{1:k}) = \mathcal{N}(x, \mu_k, \Sigma_k) \tag{6}$$

x_k is the state of the object defined as,

$$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \tag{7}$$

where x, y are the coordinates in 2D space and v_x, v_y are the respective velocities. We assume that each target evolves and generates observations independently of one another. The probability hypothesis density of multiple targets is represented by a sum of Gaussian ν_k,

$$\nu_k = \sum_{k=1}^{n} w_k \mathcal{N}(x; \mu_k, P_k) \tag{8}$$

where w_k gives the expected amount of targets at μ_k and P_k is the covariance matrix associated with each target distribution. We propagate this sum of Gaussian in the filter to obtain realizations in the PHD at each step of the recursion based of a tracking model similar to the Single Object Kalman Filter applied on the PHD instead of each multi-target state which would be intractable.

4) PHD Filter: The PHD recursion consists of 6 steps as listed below:-

1) **Target Births-** The PHD Filter was developed for aerospace applications. In such scenarios, air-crafts which have not previously observed appear in the scene from certain locations i.e air bases/airports. The PHD of these new targets can be modeled using a RFS at each time step $S_{k|k-1}$. The intensity of these new born targets is represented by a Gaussian Mixture model with intensity γ_k,

$$\gamma_k = \sum_{i}^{J_{\gamma k}} w_{\gamma k} \mathcal{N}(x; \mu_{\gamma k}, P_{\gamma k}) \tag{9}$$

$J_{\gamma k}$ is the predefined number of locations where targets are born. $\mu_{\gamma k}$,$P_{\gamma k}$ determine the properties of the birth locations and distributions. $w_{\gamma k}$ is the expected number of targets.

2) **Target Spawn**- We also account for new targets being spawned from existing targets at each time step, this

is modeled using an RFS $B_{k|k-1}$. We use this model for cases where cars appear from a blind spot. The intensity for spawned targets is modeled as a GMM $\beta_k(x|\psi)$

$$\beta_k(x|\psi) = \sum_{i}^{J_{\beta k}} w_{\beta k}\mathcal{N}(x; F_{\beta,k}\psi + d_{\beta,k}, Q_{_{\beta,k}}) \quad (10)$$

where ψ is the state of the parent target which is spawned from. $F_{\beta,k}$, $d_{\beta,k}$ and $Q_{_{\beta,k}}$ model the distribution of the spawned targets around the parent target. $w_{\beta k}$ is the expected number of targets spawned. At the end of the Birth and Spawn Process we have generated a RFS X_k,

$$X_k = S_{k|k-1} \bigcup B_{k|k-1} \quad (11)$$

3) **Existing Target Prediction**- We assign a probability p_S to the probability that a target from the previous time-step $k-1$ has survived. We use the Motion model discussed in the Kalman-Filtering section to predict the intensity at time-step k. The predicted intensity of the targets that survive to the next time step is given by $\nu_{S,k|k-1}$,

$$\nu_{S,k|k-1}(x) = p_S \sum_{j=1}^{J_k} w_{k-1}\mathcal{N}(x; \mu_{S,k-1}^{(j)}, P_{S,k-1}^{(j)}) \quad (12)$$

$$\mu_{S,k-1}^{(j)} = F\mu_{k-1}^{(j)} \qquad P_{S,k-1}^{(j)} = FP_{k-1}^{(j)}F^T \quad (13)$$

where $\mu_{S,k-1}^{(j)}$ and $P_{S,k-1}^{(j)}$ represent the distribution of the targets from the previous time-step which consists of a Gaussian Mixture of J_k targets. F is the motion model. The RFS generated by the existing targets in the next time step is denoted by Θ_k.

$$X_k = X_k \bigcup \Theta_k \quad (14)$$

The RFS X_k till step 3 is called the *predicted set of targets*.

4) **Detection and Update**- In this step use the RFS of sensor readings Z_k to compute the likelihood of each sensor reading and implement Bayes rules to compute the posterior distribution at time-step k. The intensity of the posterior distribution after the update step under the probability p_D that the target is detected is given by,

$$\nu_k(x) = (1-p_D)\nu_{S,k|k-1}(x) + \sum_{z_k \in Z} \nu_{D,k}(x; z_k) \quad (15)$$

$$\nu_{D,k}(x; z_k) = \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z)\mathcal{N}(x; \mu_{k|k}^{(j)}, P_{k|k}^{(j)}) \quad (16)$$

$$w_k^{(j)}(z) = \frac{p_D w_{k|k-1}^{(j)} q_k^{(j)}(z)}{p_D \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)} \quad (17)$$

$$q_k^{(j)}(z) = \mathcal{N}(x; H_k m_{k|k-1}, R_k + H_k P_{k|k-1}^{(j)} H_k^T) \quad (18)$$

$$\mu_{k|k}^{(j)} = \mu_{k|k-1}^{(j)} + K_k^{(j)}(z - H_k m_{k|k-1}^{(j)}) \qquad P_{k|k}^{(j)} = [I - K_k^{(j)} H_k] P_{k|k-} \quad (19)$$

$$K_k^{(j)} = P_{k|k-1}^{(j)} - H_k^T (H_k P_{k|k-1}^{(j)} H_k^T + R_k)^{-1} \quad (20)$$

$K_k^{(j)}$ is called the *Kalman Gain*, $w_k^{(j)}(z)$ and $q_k^{(j)}(z)$ are the normalized and un-normalized updated weight/likelihood of the target state after applying the Kalman Filter and sensor readings using the sensor model with parameters H_k and R_k. The posterior distribution of the target states is defined using $\mu_{k|k}^{(j)}$ and $P_{k|k}^{(j)}$. In this step we compute a belief by associating each sensor reading with every predicted target, this leads to the size of RFS increasing. However few likelihoods for measurements and targets will be very low.

5) **Pruning and Merging**- The RFS from the previous time step is Pruned and Merged to remove the estimates with low weights and merge similar measurements using a threshold on the Mahalanobis distance.
6) **Multi-target State Extraction**- Post Pruning and Merging we extract only the estimates with weights above a certain threshold value. The pseudocode for the complete algorithm is mentioned in the appendix.

IV. Results

A. Vehicle Detection and Instance Segmentation

The Mask R-CNN implementation was capable of detecting almost all cars in a given frame, and even in partially occluded cases. We observed that the classifier worked well in mildly cluttered environments, however, in densely cluttered environments, the detector missed a few instances as shown in Figure 3.

B. Dense Optical Flow

The optical flow calculated between frames gave us noisy estimates in case where the frame rate was higher than 20 fps. Estimating where the vehicle solely based on optical flow was difficult because of this. We decreased the frame rate of the images till we found the perfect compromise between noise and too low a frame rate which would hinder tracking. A frame rate of 6 fps worked the best for us. This was a qualitative observation which resulted in better tracking of the vehicles as shown in Fig.3.

C. PHD Filter

1) Simulation Result: We implemented the simulation in the paper to reproduce the results in MATLAB. The different hyper-parameters for this model can be referred from the original paper. The simulation consists of two particles born at the same time and at $k = 53s$ a third particle spawns off

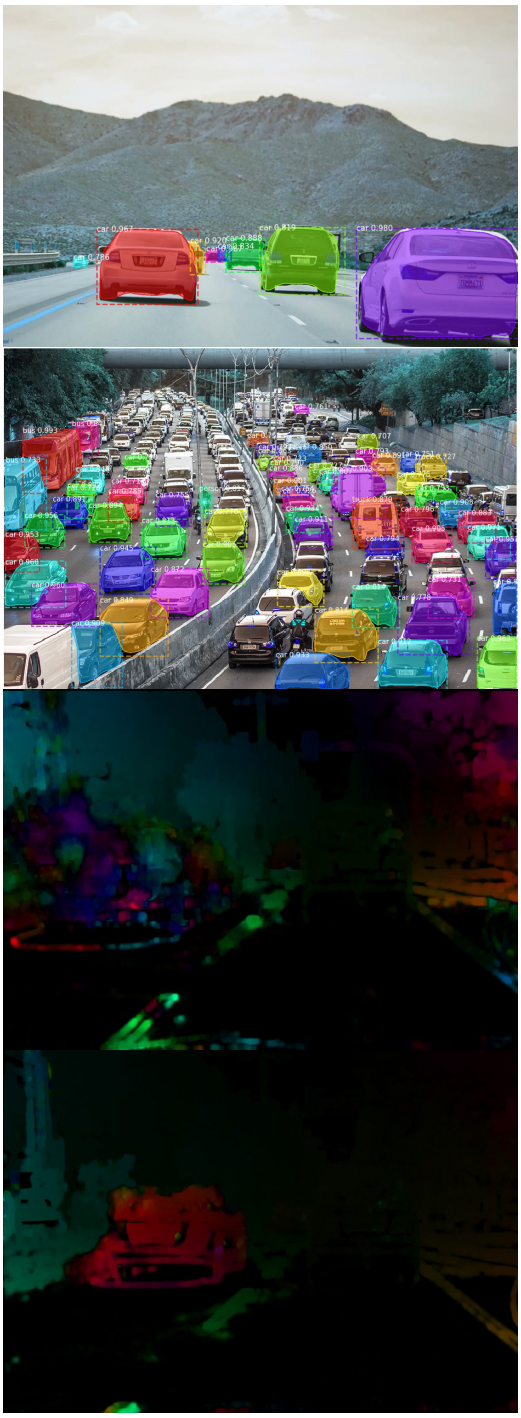

Fig. 3. *(top)*Performance of Mask R-CNN in (i) moderately dense scene , (ii) dense traffic,
*(bottom)*Optical Flow performance at: (iii) frame rates $= 20$ fps *(noisy)* (iv) frame rate $= 6$ fps

one of the particles. All three particles die at $k = 100$. At each time step k we plot the uncertainty ellipse of the target state distribution shown in Figure 4, you can view the output sequence here.

2) Car Tracking in 2D pixel space: The PHD Filter formulation along with the Mask R-CNN and Dense Optical Flow mentioned before is used to track Cars in 2D space

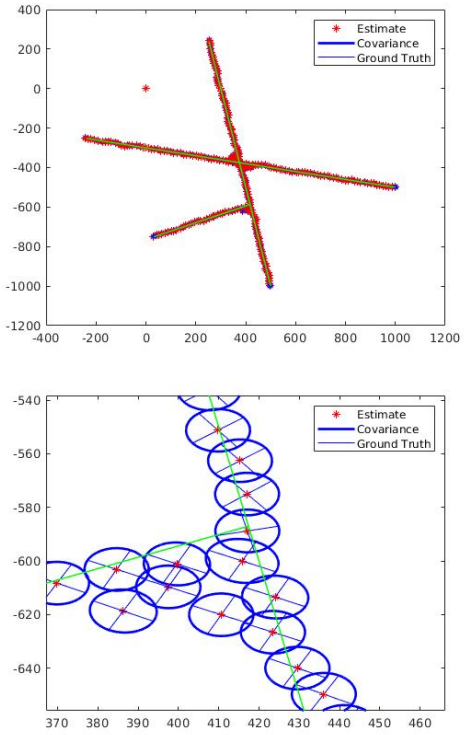

Fig. 4. *From left to right* (i) The posterior state hypothesis of two particles born at $k = 0$ and a third particle spawned at $k = 53s$, (ii) A zoomed in view of the uncertainty bounds with the mean and covariance ellipsoid about the ground truth

in the CityScape data-set. Since no ground truth exists on the location, the results we have are qualitative. What we are looking for is a good qualitative representation of the car position in image space as shown in the first two images in Figure 5, you can view the output sequence here.

We also observe the results in cluttered environments where the state estimates of different cars merge together to produce one single estimate as in the last two images in Figure 5. The implementation details for the hyper-parameters are in the original paper [13].

V. Discussion and Limitations

The Filter alleviates the combinatorial problem of data association and produces good qualitative and quantitative results. We believe the results above could be made better by further tuning the hyper parameters [13] of the filter. However the future work in this field would involve establishing a more generic framework for the filter and not over-fitting parameters for specific examples. As discussed earlier, the results by our Sensor Model are very good in most cases and when paired with a well tuned filter should give us very accurate tracking estimates. The major drawback with using Mask R-CNN is the requirement of massive computational resources with a computation time of approximately 200 ms [6] per frame on a single GPU. Moving onto faster networks, such as YOLO [10] and adding a Mask branch to it might help. Classical methods for segmentation (such as *graph cut methods*) on

Fig. 5. *(top)* (i) The posterior state hypothesis of cars $k = ts$ with the uncertainty bounds depicted by the blue ellipses, (ii) The posterior state hypothesis of cars $k = t + 1s$, *(bottom)* (iii) The posterior state hypothesis of cars $k = ts$ in which the estimates are very close (iv) The posterior state hypothesis of cars $k = t + 1s$ where the previously close estimates of the three cars have now merged together

the image windows inside the bounding boxes provided by YOLO (or other similar CNNs) may provide desirable and much faster results as well.

VI. Conclusion and Future Work

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

A. PseudoCode for PHD Recursion

Acknowledgment

This work was partially supported by the Toyota Research Institute (TRI), partly under award number N021515, however, this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

Pseudo-code for the Gaussian mixture PHD filter.

---

given $\{w_{k-1}^{(i)}, m_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^{J_{k-1}}$, and the measurement set Z_k.

step 1. (prediction for birth targets)

$i = 0$.

for $j = 1, \ldots, J_{\gamma,k}$

$i := i + 1$.

$w_{k|k-1}^{(i)} = w_{\gamma,k}^{(j)}, \quad m_{k|k-1}^{(i)} = m_{\gamma,k}^{(j)}, \quad P_{k|k-1}^{(i)} = P_{\gamma,k}^{(j)}$.

end

for $j = 1, \ldots, J_{\beta,k}$

for $\ell = 1, \ldots, J_{k-1}$

$i := i + 1$.

$w_{k|k-1}^{(i)} = w_{k-1}^{(\ell)} w_{\beta,k}^{(j)}$,

$m_{k|k-1}^{(i)} = d_{\beta,k-1}^{(j)} + F_{\beta,k-1}^{(j)} m_{k-1}^{(\ell)}$,

$P_{k|k-1}^{(i)} = Q_{\beta,k-1}^{(j)} + F_{\beta,k-1}^{(j)} P_{k-1}^{(\ell)} (F_{\beta,k-1}^{(j)})^T$.

end

end

step 2. (prediction for existing targets)

for $j = 1, \ldots, J_{k-1}$

$i := i + 1$.

$w_{k|k-1}^{(i)} = p_{S,k}\, w_{k-1}^{(j)}$,

$m_{k|k-1}^{(i)} = F_{k-1} m_{k-1}^{(j)}, \quad P_{k|k-1}^{(i)} = Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^T$,

end

$J_{k|k-1} = i$.

step 3. (construction of PHD update components)

for $j = 1, \ldots, J_{k|k-1}$

$\eta_{k|k-1}^{(j)} = H_k m_{k|k-1}^{(j)}, \quad S_k^{(j)} = R_k + H_k P_{k|k-1}^{(j)} H_k^T$,

$K_k^{(j)} = P_{k|k-1}^{(j)} H_k^T [S_k^{(j)}]^{-1}, \quad P_{k|k}^{(j)} = [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)}$.

end

step 4. (update)

for $j = 1, \ldots, J_{k|k-1}$

Pruning for the Gaussian mixture PHD filter.

---

given $\{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k}$, a truncation threshold T, a merging threshold U, and a maximum allowable number of Gaussian terms J_{max}.

Set $\ell = 0$, and $I = \{i = 1, \ldots, J_k | w_k^{(i)} > T\}$.

repeat

$\ell := \ell + 1$.

$j := \arg\max_{i \in I} w_k^{(i)}$.

$L := \left\{ i \in I \;\middle|\; (m_k^{(i)} - m_k^{(j)})^T (P_k^{(i)})^{-1} (m_k^{(i)} - m_k^{(j)}) \leq U \right\}$.

$\tilde{w}_k^{(\ell)} = \sum_{i \in L} w_k^{(i)}$.

$\tilde{m}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i \in L} w_k^{(i)} x_k^{(i)}$.

$\tilde{P}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i \in L} w_k^{(i)} (P_k^{(i)} + (\tilde{m}_k^{(\ell)} - m_k^{(i)})(\tilde{m}_k^{(\ell)} - m_k^{(i)})^T)$.

$I := I \backslash L$.

until $I = \emptyset$.

if $\ell > J_{max}$ then replace $\{\tilde{w}_k^{(i)}, \tilde{m}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{\ell}$ by those of the J_{max} Gaussians with largest weights.

output $\{\tilde{w}_k^{(i)}, \tilde{m}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{\ell}$ as pruned Gaussian components.

Fig. 7. Psuedo code for the Pruning step of the PHD Filter algorithm

MULTI-TARGET STATE EXTRACTION

given $\{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k}$.
Set $\hat{X}_k = \emptyset$.
for $i = 1, \ldots, J_k$
  if $w_k^{(i)} > 0.5$,
    for $j = 1, \ldots, \text{round}(w_k^{(i)})$
      update $\hat{X}_k := \left[\hat{X}_k, m_k^{(i)}\right]$
    end
  end
end
output $\hat{X}_k$ as the multi-target state estimate.

Fig. 8. Psuedo code for the Multi-target state extraction

REFERENCES

[1] Y. Bar-Shalom. *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[4] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, 8(3):173–184, 1983.

[5] Irwin R Goodman, Ronald P Mahler, and Hung T Nguyen. *Mathematics of data fusion*, volume 37. Springer Science & Business Media, 2013.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[7] Carine Hue, J-P Le Cadre, and Patrick Perez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Transactions on signal processing*, 50(2):309–325, 2002.

[8] EW Kamen. Multiple target tracking based on symmetric measurement equations. *IEEE Transactions on Automatic Control*, 37(3):371–374, 1992.

[9] Ronald PS Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic systems*, 39(4):1152–1178, 2003.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[12] Roy L Streit and Tod E Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. In *Signal and Data Processing of Small Targets 1994*, volume 2235, pages 394–406. International Society for Optics and Photonics, 1994.

[13] Ba-Ngu Vo and Wing-Kin Ma. The gaussian mixture probability hypothesis density filter. *IEEE Transactions on signal processing*, 54(11):4091, 2006.

[14] B. Zhang, Y. Xu, and X. Yang. Online pedestrian tracking using ensemble color feature. In *2015 IEEE International Conference*

on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pages 276–282, Oct 2015.