This is a repository copy of *A Gaussian Process Convolution Particle Filter for Multiple Extended Objects Tracking with Non-Regular Shapes*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/131615/

Version: Accepted Version

A Gaussian Process Convolution Particle Filter for Multiple Extended Objects Tracking with Non-Regular Shapes

Waqas Aftab
The University of Sheffield
Sheffield, UK
waftab1@sheffield.ac.uk

Allan De Freitas
University of Pretoria
Pretoria, South Africa
a.defreitas@tuks.co.za

Mahnaz Arvaneh
The University of Sheffield
Sheffield, UK
m.arvaneh@sheffield.ac.uk

Lyudmila Mihaylova
The University of Sheffield
Sheffield, UK
L.S.Mihaylova@sheffield.ac.uk

Abstract—Extended object tracking has become an integral part of various autonomous systems in diverse fields. Although it has been extensively studied over the past decade, many complex challenges remain in the context of extended object tracking. In this paper, a new method for tracking multiple irregularly shaped extended objects using surface measurements is proposed. The Gaussian Process Convolution Particle Filter proposed in [1], designed to track a single extended/group object, is enhanced for tracking multiple extended objects. A convolution kernel is proposed to estimate the multi-object likelihood. A target birth/death model based on the proposed method is also introduced for automatic initiation and deletion of the objects. The proposed approach is validated on real-world LiDAR data which shows that the method is efficient in tracking multiple irregularly shaped extended objects in challenging scenarios involving occlusion, dense clutter and low object detection.

I. Introduction

Recently, extended object tracking (EOT) has become a fundamental process in various autonomous systems. These systems belong to a wide spectrum of fields such as navigating autonomous cars through traffic [2], autonomous human and surrounding objects tracking based on Microsoft Kinect sensors [3], tracking of hazardous clouds [4] and many more. EOT requires estimation of the kinematics and the shape of an object of interest from a sequence of noisy sensor measurements. The kinematic states include the position of the centre of the object and its higher order time derivatives. The states related to the shape of the object are commonly the extent of the object from the centre.

Object tracking is not a new area of research and dates back to the times of the second world war. Traditionally, object tracking has been referred to as multiple target tracking (MTT) [5]. MTT deals only with the estimation of the object kinematics. It is also referred as point objects tracking (POT) [6] and [7]. The state estimation of the kinematics states in the EOT is done using the methods similar to those proposed in the traditional MTT literature [8], [9]. The focus of the EOT research has been on the shape estimation methods and the measurement models. The different methods proposed over time have been extensively covered in the two overview papers [6], [7]. Typically, the object kinematics are modelled in the global frame while the object shape is modelled in the object centered frame.

Similarly to the POT, the tracking of a single extended object is comparatively simpler to the problem of tracking multiple extended objects. A single EOT problem provides a solution to the challenges of unknown object kinematics, shape and shape dynamics, measurement error and uncertainty of the measurements origin using Bayesian inference methods. The object kinematics models are inherited from the literature of the POT. Various shape models have been proposed for the single EOT. These include techniques which model the object shape as a basic geometric shape, e.g. tracking of a cyclist using a stick model [7], a car using a rectangular model [10], a ship using an ellipsoidal model [11] etc. Although these methods have been proved to be simple and efficient on real-world data, the shapes of these and other extended objects are different to the basic geometric shapes. The tracking accuracy increases as the precision of the shape estimation increases [7]. Some advanced shape estimation methods have also been proposed for tracking the object as an irregularly shaped (star-convex) object. These include the random hypersurface model (RHM) [12], Gaussian Process (GP) based models [13], [1] and mixture of sub-objects [14].

The measurement models include the measurement source and clutter model and the sensor measurement error model. The complexity of the methods increases when the measurements are received from the surface of the object. In such scenarios, the RHM and the GP model of [13] are sensitive to the statistical properties of the measurements coming from the object, which might be unknown in real-world scenarios. The analytical expression of the measurement likelihood is also not available due to the non-linearity of the problem. The Gaussian Process Convolution Particle Filter (GPCPF) [1] does not require any prior knowledge of measurement statistics and an analytical expression of the likelihood function. Moreover, unlike the method proposed in [13], this method can track a single object moving in clutter.

The multiple EOT provides a solution to the unknown number of objects and the data association problem in addition to the problems of the single EOT. To solve the problem in a reasonable amount of time using the computational power, the process of clustering measurements is done prior to the multi-object state estimation. The estimation accuracy is therefore sensitive to the clustering technique and a lot of research is done in measurement partitioning methods. The DBSCAN [15] based clustering method provides good

performance for classification of multiple objects. However, advanced clustering and inference techniques are required when the objects come close or cross each other. In such scenarios, the most likely sets of measurement clusters are used for inference. A stochastic optimisation based method [16] has been demonstrated to track closely moving objects by proposing an efficient method to determine the most likely sets. In this paper, a GPCPF based method is proposed for tracking of multiple extended objects with non-regular shapes.

A. Contributions

The contributions of this work are as follows; (i) A new Gaussian process convolution particle filter (GPCPF) based approach for tracking multiple extended objects having non-regular shapes is proposed. A GPCPF for tracking a single extended object is proposed in [1]. (ii) A new convolutional kernel is proposed to track different complex shaped objects using surface measurements without any prior knowledge of the measurement statistics. The typical complex-shaped multiple extended objects tracking methods require prior information of the object size or the statistical properties of the measurements [17], [18]. (iii) A new object birth/death model based on the likelihood estimation using convolution kernel is proposed. This framework treats the object detection, false-alarm rejection, object existence and death in a probabilistic framework without the requirement of an explicit likelihood function. (iv) The performance validation of the proposed method on real data from extended objects is presented in the results section.

The structure of the paper is as follows. The system dynamical model is presented in Section II, the theoretical background of GP and GPCPF is described in Section III, the proposed multiple EOT GPCPF is given in Section IV, the performance validation and results are given in Section V followed by conclusions in Section VI.

II. SYSTEM MODEL

A. System Dynamics Model

The dynamics of the centre of the object (COO) are assumed independent of those of the object shape. The discrete time COO state update equation is given below:

$$c_k = (I_{N_{g,k}} \otimes F^c)c_{k-1} + w^c_{k-1}, \tag{1}$$

where $c_k = \left[\left(c_k^1\right)^T, \left(c_k^2\right)^T, \cdots, \left(c_k^{N_{g,k}}\right)^T \right]^T$ represents the multiple objects COO state vector, I_n represents an n-dimensional identity matrix, $(\cdot)^T$ represents the transpose operation, $N_{g,k}$ represents the number of extended objects at time k, F^c represents the single object COO state transition matrix, $w^c_{k-1} \sim \mathcal{N}\left(0, I_{N_{g,k}} \otimes Q^c\right)$ represents the COO model process noise, Q^c represents the process noise covariance of the COO of a single object and $(.)_k$ represents that the vector / matrix corresponds to time k. The extent states dynamics is modelled as a random walk [1] and is described by the following equation:

$$s_k = (I_{N_{g,k}} \otimes I_B)s_{k-1} + w^s_{k-1}, \tag{2}$$

where $s_k = \left[\left(s_k^1\right)^T, \left(s_k^2\right)^T, \cdots, \left(s_k^{N_{g,k}}\right)^T \right]^T$ represents the multiple objects extent state vector, $w^s_{k-1} \sim \mathcal{N}\left(0, I_{N_{g,k}} \otimes Q^s\right)$ represents the extent dynamics model noise, Q^s repre-

sents the process noise covariance of a single object extent dynamics. The Q^s can be modelled based on the prior knowledge of the objects being tracked, e.g. if the objects are axis-symmetric then an axis-symmetric covariance kernel can be used to determine this matrix. If there is no prior knowledge of the object shape, then it can be modelled as given below [1]:

$$Q^s = \sigma_e^2 I_B, \tag{3}$$

where σ_e^2 represents the variance of the change in extent per sample time.

B. State Vector

The multiple objects state vector X_k is given below:

$$X_k = \left[(x_k^1)^T \quad (x_k^2)^T \quad \cdots \quad (x_k^{N_{g,k}})^T \right]^T, \tag{4}$$

$$x_k^i = \left[(c_k^i)^T \quad (s_k^i)^T \right]^T, \tag{5}$$

where x_k^i represents the ith object state, c_k^i represents the states related to the centre and s_k^i denotes the states related to the shape (extent) of the ith object. The extent states consist of the radial extent of the object at B different angles from the COO. The COO and the extent states are given below:

$$c_k^i = \left[x_k^i \quad \dot{x}_k^i \quad y_k^i \quad \dot{y}_k^i \right]^T, \tag{6}$$

$$s_k^i = \left[r_k^{i,1} \quad r_k^{i,2} \quad \cdots \quad r_k^{i,B} \right]^T. \tag{7}$$

where (x_k^i, y_k^i) and $(\dot{x}_k^i, \dot{y}_k^i)$ represent the position and velocity of the ith object's COO. $r_k^{i,j}$ represents the radial value of the ith object at the jth angle of the input vector θ^b. The input angle vector is given below:

$$\theta^b = \left[\theta^1 \quad \theta^2 \quad \cdots \quad \theta^B \right]^T, \quad \theta^l = (l-1)\frac{2\pi}{B}. \tag{8}$$

C. Shape Model and the GP

The shape of the object is assumed to be star-convex[1] and is modelled using a Gaussian Process (GP) model as proposed in [13]. The extent is modelled as a function of the angle from the COO. This mapping function for an ith object is given by the following equation:

$$r^i = f^i(\theta), \tag{9}$$

where r^i represents the true radial values and f^i represents the true mapping function of the ith object. The function maps the continuous domain θ to the ranges r. As the object can have any arbitrary shape hence f^i is a non-linear function and a GP is used to estimate this mapping function. This is further realised in Figs 1a and 1b. A Von Mises covariance kernel [1] is used:

$$k_{vm}(\theta_i, \theta_j') = \sigma_f^2 e^{\sigma_a^2 cos(\theta_i - \theta_j')}, \tag{10}$$

where σ_f^2, σ_a^2 control the magnitude and the length-scale of the kernel, respectively.

D. Measurement Model

The sensor reports measurements in Cartesian coordinates and the measurement noise is assumed to be independent and identically distributed Gaussian. The measurement equation is given below:

$$z_k = \mathcal{H}(X_k) + v_k, \quad v_k \sim \mathcal{N}(0, \Sigma_k), \tag{11}$$

$$\Sigma_k = diag(R_{1,k}, R_{2,k}, \cdots, R_{N_k,k}), \tag{12}$$

---

[1]A polygon is called star-convex, if all line segments from its center to the boundary lie inside the same polygon.

(a) Extended object model   (b) Visualisation of $f^{i,p}$
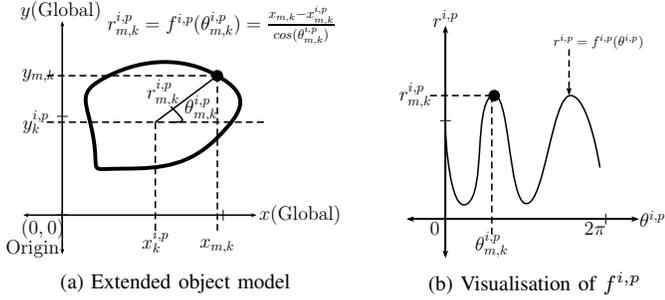
Fig. 1: (a) Fig. 1a shows an ith extended object (thick solid line) in the global Cartesian frame. The sensor measurements and the COO kinematics are modelled in the global frame. The extent states are modelled in the polar frame local to each object. The ith object's local frame has origin located at $(x_k^{i,p}, y_k^{i,p})$. The radial extent r of the object is modelled as a function f of the angle θ in the local frame given by $r = f(\theta)$. The coordinates of the mth measurement are shown in the global and the ith objects local frame. The non-linear relation between the two frames is also presented for this measurement. (b) Fig. 1b visualises the non-linear radial function f of Fig. 1a. The origin corresponds to the centre of the ith object. The mth measurement is shown for comparison.

where $z_k = [z_{1,k}^T, z_{2,k}^T, \cdots, z_{N_k,k}^T]^T$ denotes the measurement vector, N_k represents the total number of measurements, $\mathcal{H}(\cdot)$ represents the non-linear measurement function, v_k is the measurement noise vector, Σ_k is the measurement noise covariance matrix, $z_{m,k} = [x_{m,k}, y_{m,k}]^T$ represents the mth measurement, $(x_{m,k}, y_{m,k})$ are the Cartesian coordinates of the mth measurement, $R_{m,k} = diag(\sigma_{x_m}^2, \sigma_{y_m}^2)$ is the corresponding measurement noise covariance matrix, $\sigma_{x_m}^2, \sigma_{y_m}^2$ represent the sensor noise variances in the x and y dimensions, respectively and $diag(\cdot)$ represents a diagonal matrix.

III. THEORETICAL BACKGROUND

A. Gaussian Process

A GP is a stochastic process for mapping non-linear functions from an input to an output space. The GP is defined by a mean and a covariance kernel. The parameters of the mean and the covariance kernel are called hyperparameters. These hyperparameters are learned using training data. A trained and learned GP can be used to determine the output at input locations other than those present in the training data. An elaborate account of the GP and its applications can be found in [19].

Let the input and output spaces be represented by the random vectors θ and r, respectively. A GP, mathematically represented as $GP(\mu(\theta), C(\theta, \theta'))$, is described by a non-linear function f given below:

$$r = f(\theta), \tag{13}$$

$$C(\theta, \theta') = \begin{bmatrix} k(\theta_1, \theta_1') & k(\theta_1, \theta_2') & ... & k(\theta_1, \theta_N') \\ k(\theta_2, \theta_1') & k(\theta_2, \theta_2') & ... & k(\theta_2, \theta_N') \\ \vdots & \vdots & \vdots & \vdots \\ k(\theta_N, \theta_1') & k(\theta_N, \theta_2') & ... & k(\theta_N, \theta_N') \end{bmatrix}, \tag{14}$$

where the mean of the GP is generally modelled as a constant, $C(\theta, \theta')$ is the covariance matrix of the GP and $k(\cdot, \cdot)$ is the

TABLE I: GP-CPF Recursion

1	for $k \leq 2$, $\theta_B = 0 : 360/B : 360$ Initialise state x_0
2	for $k = 3$ find $\tilde{x}_0^n = \mathcal{N}(x_0, \sigma_{x_0}^2)$, $w_0^n = \frac{1}{N}$
3	for $k > 3$ Re-sample : Residual Re-sampling as in [20].
4	for $k \geq 3$
4a	State Sample: for $n = 1, 2, ...N$
	$\tilde{x}_k^n = F\tilde{x}_{k-1}^n + w_k$
4b	Measurement Simulation : Simulate measurements using the measurement model
4c	Measurement Gating : A measurement is considered gated if it lies within the region of the measurement sample
4d	Weight Update : for $n = 1, 2, ...N$ find
	$w_k^n = \prod_{l=1}^{L} K_h^z(z_{k,l}) \times w_{k-1}^n$
4e	Normalise Weight : for $n = 1, 2, ...N$ determine
	$w_k^n = \frac{w_k^n}{\sum_{n=1}^{N} w_k^n}$
4f	Estimation : $p_k^n(x_k|z_{1:K}) = \sum_{n=1}^{N} w_k^n \tilde{x}_k^n$

corresponding covariance kernel. Let $\theta' \in \mathbb{R}^N$ and r' represent the training data input and output vectors, respectively. The GP regression equation for an unknown input vector $\theta^\star$ is given below:

$$p(r^\star|\theta^\star) = \mathcal{N}(C_{\theta^\star\theta'}(C_{\theta'\theta'} + \sigma^2 I_N)^{-1}r', C_{\theta^\star\theta^\star}$$
$$- C_{\theta^\star\theta'}(C_{\theta'\theta'} + \sigma^2 I_N)^{-1}C_{\theta'\theta^\star}), \tag{15}$$

where $r^\star$ represents the output vector, $(\cdot)^{-1}$ represents the matrix inverse and σ^2 is measurement noise variance.

B. Gaussian Process Convolution Particle Filter

The GPCPF [1] tracks an irregularly shaped extended object moving through clutter using noisy sensor measurements originating from the surface of an extended object. The filter has two major components namely the GP model for the object shape and the CPF for the posterior density estimation. The GPCPF recursion is summarised in Table I.

IV. MULTIPLE GROUP OBJECTS TRACKING USING GAUSSIAN PROCESS CONVOLUTION PARTICLE FILTER

The multivariate density of the multiple extended objects states is multimodal. Each mode of the density corresponds to an extended object in the real-world. The number of objects and hence the number of modes is assumed unknown. Furthermore, the measurements origin and the object kinematics (both of the COO and the extent) are also assumed unknown. The sensor data is assumed to be noisy, gives multiple surface measurements per extended object and reports measurements from the extended objects as well as clutter. The existence, birth and death probabilities of the objects are also unknown.

The Gaussian Process Convolution Particle Filter (GPCPF) [1] has been shown to be able to track a single irregularly shaped extended object moving through clutter. In this work, the GPCPF is extended to track multiple extended objects from measurements with clutter. The multivariate density function is determined using the system dynamics defined by equations (1) and (2). The samples are drawn from (1) and (2) assuming Gaussian process noise. As a result, the proposed density estimation is performed using a mixture of Gaussians. These (Gaussian) samples are mapped as irregularly shaped regions in the measurement space after the measurement sampling is done. The sensor measurements

are then used to update the posterior density estimation using the CPF measurement kernel.

A. The Convolution Particle Filter Kernel

The CPF kernel is used for the multi-modal density estimation. The CPF in a state space setting for the multiple objects tracking relies on two kernels. The first kernel $K^{\boldsymbol{X}}_{h_{\boldsymbol{X}}}$ is defined for the predictive distribution in the state space of state vector $\boldsymbol{X}$. The second kernel is defined for the likelihood estimation in the measurement space $K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}$ of the measurement vector $\boldsymbol{z}$. The $h_{\boldsymbol{X}}$ and $h_{\boldsymbol{z}}$ represent the predictive and the likelihood kernel bandwidths, respectively. As proposed in [1], the state vector $\boldsymbol{X}_k$ maps to multiple independent regions in the measurement space and is equivalent to the functionality of the kernel $K^{\boldsymbol{X}}_{h_{\boldsymbol{X}}}$. Hence, $K^{\boldsymbol{X}}_{h_{\boldsymbol{X}}}$ is not required to be defined explicitly. The likelihood kernel $K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}$ is defined for measurements originating from the objects as well as clutter. This kernel has the form:

$$K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{z}^m - \tilde{\boldsymbol{Z}}^{i,p}_k) = \begin{cases} \mathcal{U}_{\mathcal{Z}^{i,p}}(\boldsymbol{z}), & \boldsymbol{z}^m \in \mathcal{Z}^{i,p} \\ \mathcal{U}_{\mathcal{V}}(\boldsymbol{z}), & \boldsymbol{z}^m \notin \mathcal{Z}^{i,p} \end{cases}, \quad (16)$$

where $\boldsymbol{z}^m$ represents the mth measurement, $\tilde{\boldsymbol{Z}}^{i,p}_k$ represents the measurement sample of the pth particle and the ith object, $\mathcal{U}_{\mathcal{R}}$ represents a uniform distribution supported by the region $\mathcal{R}$, $\mathcal{V}$ represents the sensor coverage and $\mathcal{Z}^{i,p}$ represents the ith region in the measurement space of the pth particle. Each particle creates $N_{g,k}$ regions in the measurement space. The uniform kernels described in (16) are given below:

$$\mathcal{U}_{\mathcal{Z}^{i,p}}(\boldsymbol{z}) = \frac{1}{Area(\mathcal{Z}^{i,p})}, \quad \mathcal{U}_{\mathcal{V}}(\boldsymbol{z}) = \frac{1}{Area(\mathcal{V})}, \quad (17)$$

where $Area(\cdot)$ returns the area of the region within brackets.

B. State Sampling

The state sample of a pth particle at time k is given below:

$$\tilde{\boldsymbol{X}}^p_k = \left[(\tilde{\boldsymbol{x}}^{1,p}_k)^T \quad (\tilde{\boldsymbol{x}}^{2,p}_k)^T \quad \cdots (\tilde{\boldsymbol{x}}^{N_{g,k},p}_k)^T \right]^T, \quad (18)$$

$$\tilde{\boldsymbol{x}}^{i,p}_k = \left[(\tilde{\boldsymbol{c}}^{i,p}_k)^T \quad (\tilde{\boldsymbol{s}}^{i,p}_k)^T \right]^T, \quad (19)$$

where $\tilde{\boldsymbol{X}}^p_k$ represents the multiple objects state sample, $\tilde{\boldsymbol{x}}^{i,p}_k$ represents the ith object state and $\tilde{\boldsymbol{c}}^{i,p}_k$ and $\tilde{\boldsymbol{s}}^{i,p}_k$ represent the ith object COO and the extent states sample of the pth particle. The COO and the extent states samples are determined using equations (1) and (2), respectively.

C. Measurement Sampling

The measurement sample is given below:

$$\tilde{\boldsymbol{Z}}^{i,p}_k = \left[(\tilde{\boldsymbol{z}}^{i,p}_{1,k})^T \quad (\tilde{\boldsymbol{z}}^{i,p}_{2,k})^T \quad \cdots \quad (\tilde{\boldsymbol{z}}^{i,p}_{B,k})^T \right]^T, \quad (20)$$

where $\tilde{\boldsymbol{z}}^{i,p}_{b,k}$ represents the measurement sample of the bth extent state and is determined as follows:

$$\tilde{\boldsymbol{z}}^{i,p}_{b,k} = [\tilde{x}^{i,p}_k, \tilde{y}^{i,p}_k]^T + [\cos\theta^b, \sin\theta^b]^T \odot \tilde{\boldsymbol{s}}^{i,p}_{b,k} + \boldsymbol{v}_{b,k}, \quad (21)$$

$$\boldsymbol{v}_{b,k} \sim \mathcal{N}\left(0, diag(\sigma^2_x, \sigma^2_y)\right) \quad ,$$

where $(\tilde{x}^{i,p}_k, \tilde{y}^{i,p}_k)$ represent the positional samples of the $\boldsymbol{c}^{i,p}_k$, $\tilde{\boldsymbol{s}}^{i,p}_{b,k}$ represents the bth extent state sample of $\boldsymbol{s}^{i,p}_k$, $\odot$ represents element-wise product, $\boldsymbol{v}_{b,k}$ represents the sensor noise and σ^2_x, σ^2_y represent the sensor noise variances, respectively. The measurement sample $\tilde{\boldsymbol{Z}}^{i,p}_k$ is a collection of points in the measurement space. These points are used to train the GP of the ith object and the pth particle. The GP regression (15)

is then used to define a region in the measurement space for the ith object and the pth particle denoted as $\mathcal{Z}^{i,p}$.

D. Likelihood Calculation / Weight Update

Consider N_k measurements are received at time k from the sensor in Cartesian coordinates. To perform the likelihood calculation and the weight update, the measurements are first gated with the particle measurement samples. This gating is done in two steps. First, the measurements are gated based on their locations and subsequently, the measurement clustering information is included to improve the gating process.

As soon as the measurements are received, the measurement vector $\boldsymbol{z}_k$ is clustered using DBSCAN [15]. For each measurement m the polar coordinates are determined as given below:

$$\theta^{i,p}_{m,k} = tan^{-1}\left(\frac{y_{m,k} - \tilde{y}^{i,p}_k}{x_{m,k} - \tilde{x}^{i,p}_k} \right), \quad (22)$$

$$r^{i,p}_{m,k} = \sqrt{(x_{m,k} - \tilde{x}^{i,p}_k)^2 + (y_{m,k} - \tilde{y}^{i,p}_k)^2}, \quad (23)$$

where $(\theta^{i,p}_{m,k}, r^{i,p}_{m,k})$ represents the polar coordinates of the mth measurement in the local frame of the ith object and the pth particle. The GP is used to predict the range of the ith extended object of the pth particle at an angle $\theta^{i,p}_{m,k}$ as given below:

$$\tilde{r}^{i,p}_{m,k} = \boldsymbol{C}_{\theta^{i,p}_{m,k}\boldsymbol{\theta}^b} \boldsymbol{C}^{-1}_{\boldsymbol{\theta}^b\boldsymbol{\theta}^b} \tilde{\boldsymbol{s}}^{i,p}_k, \quad (24)$$

The measurement is considered belonging to the ith extended object of the pth particle if $r^{i,p}_{m,k} \leq \tilde{r}^{i,p}_{m,k}$. The cluster identifier vector $\boldsymbol{z}_c$ of all the gated measurements is formed. The gated measurements are declared not-gated with the ith extended object of the pth particle if the cluster identifier is different from the mode of $\boldsymbol{z}_c$ or more than 15% of the measurements with same cluster identifier are not gated. The particle weights update equation is as follows:

$$w^{i,p}_k = w^{i,p}_{k-1} \prod_{m=1}^{N_k} K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{z}^m - \tilde{\boldsymbol{Z}}^{i,p}_k), \quad (25)$$

where $w^{i,p}_k$ represents the weight of the ith object and the pth particle. The measurements gated with one object are not considered for updating the other objects.

E. Estimation

The conditional multi-object state density can be written as:

$$p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}) = \frac{p(\boldsymbol{X}_k, \boldsymbol{Z}_{1:k})}{\int p(\boldsymbol{X}_k, \boldsymbol{Z}_{1:k})d\boldsymbol{X}_k}, \quad (26)$$

where $\boldsymbol{Z}_{1:k}$ represents all the measurements from time-step 1 to k. Along the lines of adaptive CPF modelled in [21], the kinematic and extent states are sampled separately. The estimate equations are given below:

$$p^P_k(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}) = \sum_{i=1}^{N_{g,k}} p^P_k(\boldsymbol{x}^i_k|\boldsymbol{Z}_{1:k}), \quad (27)$$

$$p^P_k(\boldsymbol{x}^i_k|\boldsymbol{Z}_{1:k}) = \frac{\sum_{p=1}^{P} w^{i,p}_k \tilde{\boldsymbol{x}}^{i,p}_k K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{Z}_{1:k} - \tilde{\boldsymbol{Z}}^p_{i,k})}{\sum_{p=1}^{P} K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{Z}_{1:k} - \tilde{\boldsymbol{Z}}^p_{i,k})}, \quad (28)$$

and the kernel is represented as:

$$K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{Z}_{1:k} - \tilde{\boldsymbol{Z}}^p_{i,k}) = \prod_{j=1}^{k} K^{\boldsymbol{z}}_{h_{\boldsymbol{z}}}(\boldsymbol{z}_j - \tilde{\boldsymbol{Z}}^p_{i,k}), \quad (29)$$

TABLE II: Existence processes

Process	$e_{g,k}$	$e_{g,k-1}$
Pre-birth	0	0
Birth	1	0
Existing	1	1
Death	0	1
False alarm	2	0

where P is the number of particles. The ith object state estimate is given below

$$\hat{\boldsymbol{x}}_k^i = \frac{\sum_{p=1}^{P} w_k^{i,p} \boldsymbol{x}_k^{i,p}}{\sum_{p=1}^{P} w_k^{i,p}}. \tag{30}$$

F. Object Existence / Birth / Death Model

The objects enter, pass-through and leave the area of interest. The sensors can also report clutter. These are represented by different processes which are a modification of the method proposed in [22]. The entry is modelled by a pre-birth and birth process, the pass-through is modelled by an existence process while exiting is modelled by a disappearance/death process. The sensor clutter is modelled as a false alarm process. Each extended object state is augmented by an existence variable $e_{g,k} \in \{0, 1, 2\}$ which specifies the existence state of the gth extended object at time k. The relation between the different processes and the existence variable is shown in Table II. The existence variable is assigned a value based on the object likelihood $\lambda_{g,k}$ and is given below:

$$\lambda_{g,k} = \frac{\sum_{p=1}^{P} w_k^p \prod_{m=1}^{M} K_{h_{\boldsymbol{z}}}^{\boldsymbol{z}}(\boldsymbol{z}^m - \bar{\boldsymbol{z}}_k^{g,p})}{\sum_{p=1}^{P} w_k^p}. \tag{31}$$

Two thresholds are defined to detect the object process. These are the birth threshold T_b and the death threshold T_d. The thresholds are related to the existence variable as given below:

$$e_{g,k} = \begin{cases} 1 & \lambda_{g,k} \geq T_b, e_{g,k-1} = 0 \\ 0 & \lambda_{g,k} < T_b, e_{g,k-1} = 0 \\ 0 & \lambda_{g,k} \leq T_d, e_{g,k-1} = 1 \\ 1 & \lambda_{g,k} > T_d, e_{g,k-1} = 1 \\ 2 & \lambda_{g,k} \leq T_d, e_{g,k-1} = 0 \end{cases} \tag{32}$$

At any given time, the pre-birth, birth and the existing objects are part of the extended object state vector $\boldsymbol{X}_k$. The death and false-alarm objects are removed from this state vector at the end of the processing step. As a result, the size of this state vector changes, which is also depicted by the time-dependence of $N_{g,k}$.

The objects can appear from a region called birth region e.g it can be a door to the building entrance in a crowd tracking in a building problem. There are N_b number of birth regions in the area of interest. Each birth region is defined by a centre (x_b, y_b), which specifies the location of the centre of the birth objects, an initial velocity $(\dot{x}_b, \dot{y}_b)$ and a circular region of radius r_b, which specifies the initial shape of the object. The values of these parameters can be tuned according to the application.

G. Object Merging / Splitting / Spawning

The object merging occurs by design through the gating process. The splitting/spawning can be included through a modification of the birth process. All the un-associated measurements in a particular scan are classified using DBSCAN clustering method. All the clusters are considered as birth regions for the next scan. The mean and the variance of the measurements position are used to define the centre and the size of the birth region. As a result, the object splitting/spawning is achieved.

V. PERFORMANCE VALIDATION

A sample of publicly available real-world data is considered for the performance validation [23]. This is a recorded data of the sensors installed on a car for real-world computer vision benchmarking. The benchmarking problems are related to an autonomous vehicle project. Multiple sensors data, installed on an observer vehicle, is available for various scenarios. The sensors include two gray-scale cameras, two colour cameras and one laser scanner. The data of the laser scanner (HDL-64E LiDAR) sensor is considered for the performance evaluation of the proposed approach. The ground truth data is not available and is constructed manually using the image data from one of the colour cameras. The ground truth of the states is calculated only for those time samples when the complete object is visible in the image data.

The 3D LiDAR data is reported in the body-fixed frame of the observer vehicle. The data is synchronised with the images obtained from the cameras. A 3D to 2D transformation matrix is used to project the data on the 2D image frame. The EOT is done in the image frame and compared with the ground truth data, which is also available in the image frame.

The given data sample consists of static and moving extended objects. The moving objects are considered as objects of interest for the performance validation. Hence, the static extended objects are treated as clutter. Four moving objects (cars) cross in front of the observer vehicle during a total of 66 time samples. These are available in the scene at different time instants which are explained next. The first object in samples $k = 1 - 8$, the second in $k = 1 - 22$, the third in $k = 18 - 43$ and the fourth in $k = 35 - 60$. The time samples when the objects are completely visible are as follows. The first object in samples $k = 1 - 3$, the second object in $k = 3 - 20$, the third in $k = 23 - 39$ and the fourth in $k = 39 - 55$.

The different challenges in the data are a large number of the LiDAR data i.e. on average 0.1 million measurements are received per time sample, dense (static) clutter, occlusion and one of the objects is not perfectly detected by the sensor i.e. it is a stealthy object. This stealthy object poses an additional challenge of tracking similar extended objects having different measurement statistics.

The evaluation of the multiple objects state is done using the mean cardinality $card_k^{\mu}$ comparison, the positional and velocity root mean square errors (RMSE) of the COO and the mean shape precision and recall in 200 Monte Carlo runs. The shape recall and precision has been used in computer vision for evaluating rectangular objects detection performance [24]. The RMSE errors and the shape recall and precision are calculated

as given below:

$$\hat{E}_k^a = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} (a_k^i - \hat{a}_k^i)^2, \tag{33}$$

$$R_k^\mu = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{Area(T_k^i \cap E_k^i)}{Area(T_j^i)}, \tag{34}$$

$$P_k^\mu = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{Area(T_k^i \cap E_k^i)}{Area(E_k^i)}, \tag{35}$$

where $\hat{E}_k^a$ represents the RMSE of the evaluation parameter a at time k, a_k^i represents the true and $\hat{a}_j^i$ represents the estimated value, R_k^μ and P_k^μ represent the mean shape recall and precision at time k, T_k^i represents the true shape, E_k^i represents the estimated shape, $\cap$ represents the intersection of two star-convex polygons and $Area(p)$ represents the area of the polygon p.

A. System Dynamics

The COO dynamics are modelled using a nearly constant velocity (NCV) motion model as given below:

$$\boldsymbol{F^c} = diag(\boldsymbol{F'}, \boldsymbol{F'}), \quad \boldsymbol{F'} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \tag{36}$$

$$\boldsymbol{Q^c} = diag(\sigma_{v_x}^2 \boldsymbol{Q'}, \sigma_{v_y}^2 \boldsymbol{Q'}), \quad \boldsymbol{Q'} = \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^2}{2} & \Delta T \end{bmatrix}, \tag{37}$$

where $\boldsymbol{F'}$ and $\boldsymbol{Q'}$ represent the state transition matrix and the process noise covariance matrix in one dimension, respectively, ΔT represents the sampling time, $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ represent the variances of the COO velocities. The extent process noise covariance is modelled using a periodic covariance kernel [19] and is given below:

$$\boldsymbol{Q^s} = \boldsymbol{C}_{\boldsymbol{\theta^b \theta^b}}^{per}, \quad k_\theta^{per}(\theta, \theta') = \sigma_f^2 e^{-\frac{2sin^2\left(\frac{\theta - \theta'}{2}\right)}{l_\theta^2}}, \tag{38}$$

where $\boldsymbol{C}_{\boldsymbol{\theta^b \theta^b}}^{per}$ represents a GP covariance matrix calculated using a periodic covariance kernel $k_\theta^{per}(\theta, \theta')$ and (14), σ_f^2 represents the magnitude and l_θ^2 represents the lengthscale hyperparameter.

B. Birth/Death Model

The objective is to track the moving objects of interest i.e. pedestrians, cyclists, vehicles etc. The birth/death model is enhanced based on the problem at hand. In order to detect and track only the moving objects, two speed thresholds are introduced in the birth/death model. These are the low speed threshold V_l and high speed threshold V_h. The objects of interest move with speeds higher than V_l and lower than V_h.

C. Measurement Clustering

The LiDAR data is in 3D. The measurements coming from the ground and from very high objects (which cannot be considered moving objects on the roads) are filtered based on the height information. The filtered data is clustered based on the depth value using 1D DBSCAN clustering. The measurements are then projected to the 2D image frame. The projected measurements are clustered using 2D DBSCAN clustering.

D. Parameters

The filter parameters are given as follows. The total number of time samples are $K = 66$, the sampling time is $\Delta T = 0.1s$,

the velocity standard deviations are $\sigma_{v_x} = 250p/s^2$ and $\sigma_{v_y} = 25p/s$, the hyperparameters of the extent process noise covariance kernel are $\sigma_f^2 = 10$ and $l_\theta = 0.2$, the surveillance volume is $Area(\mathcal{V}) = 1242p \times 345p$. The hyperparameters of the GPCPF kernel are $\sigma_a^2 = \frac{1}{40}$, $\sigma_r^2 = 1$ and $\sigma_f^2 = 30$. The number of particles is $N = 500$, number of basis is $B = 36$, the birth threshold is $T_b = 0.01$, death threshold is 0.001, the low speed threshold is $V_l = 200p/s$ and the high speed threshold is $V_h = 1000p/s$. The 1D DBSCAN clustering parameters are $epsilon = 1.25$ and the minimum number of points are 24. The 2D DBSCAN clustering parameters are $epsilon = 50$ and the minimum number of points are 80. The sensor noise variances are $\sigma_x^2 = \sigma_y^2 = 0.0025$.

E. Results

The challenging scenarios and the tracking results at three chosen time samples are shown in Fig 2. The statistical properties of the sensor measurements coming from object 3 (black car), shown in Fig. 2c, are different from those of the other three objects. The measurement density is different from the other similar objects. The proposed algorithm detects and tracks this object, which shows that the proposed method is not sensitive to the statistical properties of the sensor measurements.

The mean cardinality results are shown in Fig. 3. A delay in the object detection can be observed for all four objects. This is due to the fact that the shape is not detected in the initial time steps as the complete object is not visible. Moreover, a moving object can be determined from the data of minimum 2 time samples. An error in the cardinality is observed between samples $20 - 25$. It is due to the fact that the measurement statistics of the black car change considerably during these time samples. A large number of particles can be used to improve the cardinality at these time samples at the expense of computational time. The average state estimate errors are shown in Fig. 4. The positional RMSE in x is less than $25p$, y is less than $7p$, $\dot{x}$ is less than $110p/s$ and $\dot{y}$ is less than $30p/s$. The mean shape recall is greater than 0.9 for most of the time steps, which shows that more than 90% of the true shape has been recalled all the time. The mean shape precision is more than 0.8 most of the time which shows that less than 20% of the estimated shape is false.

The program was run on MATLAB R2016b and a Windows 10 (64 bit) Desktop computer installed with an Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz(4 CPUs) and 8GB RAM. The computational time is $52s$ per time sample for $N = 500$ particles. This is due to the large number of sensor measurements received at each time sample. The processing time is improved to $4.5s$ using $N = 50$ particles. The mean cardinality and the state estimate errors for $N = 50$ are also given in the Figs 3 and 4, respectively. It can be observed that all four objects are tracked and there is no false alarm. The performance of the state estimates is almost similar and the cardinality estimates are slightly degraded. The processing time can be further improved by optimizing the code and running in C++.

---

[2]p represents pixel.

(a) **Multiple extended objects**

(b) **Occluded Object**

(c) **Stealthy Object**
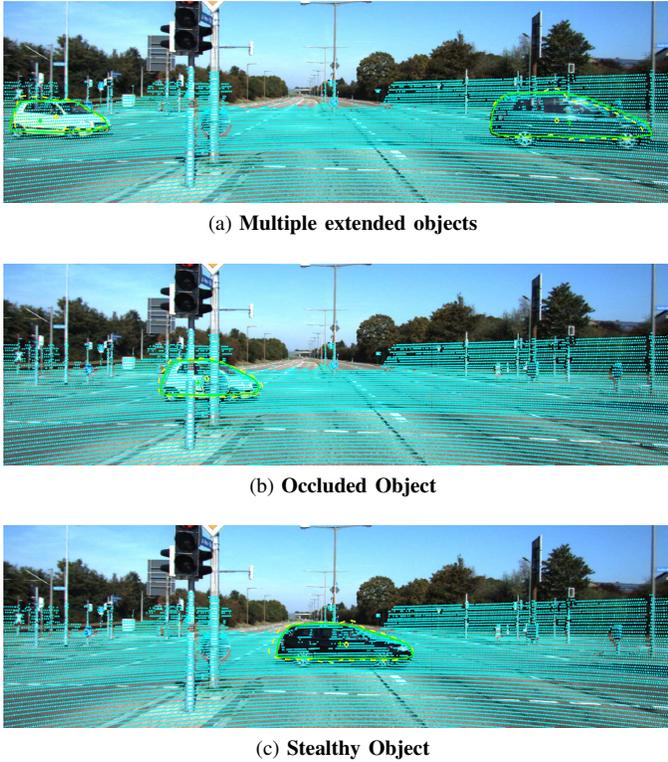
Fig. 2: **Challenging scenarios**. The projected LiDAR data (cyan dots) is overlaid on the camera image. The ground truth object is represented by the (green) solid line and the estimated object is represented by (yellow) dotted lines. The ground truth COO is represented by (green) plus and the estimated COO is represented by (yellow) diamond. (a) Two moving extended objects (cars) are tracked whereas the static extended objects (signal post, trees etc.) are treated as clutter. (b) The moving object (white car) is occluded by two static extended objects. (c) The front half of the car is picked up by the sensor while few measurements are reported from the back half of the car. The statistical properties of the sensor measurements are also different from the other moving and static extended objects.

VI. Conclusions

The paper proposes a novel GPCPF based approach for tracking irregularly shaped multiple extended objects moving through clutter. The GPCPF along with measurement clustering track the extended objects as a mixture of Gaussian state samples and measurement simulations. The performance evaluation of the approach is done on real-world data. The proposed filter is able to track non-regular shaped objects in challenging scenarios like dense clutter, occlusion and low detection. In future, the GPCPF will be enhanced to the tracking scenarios involving closely moving irregularly shaped extended objects.
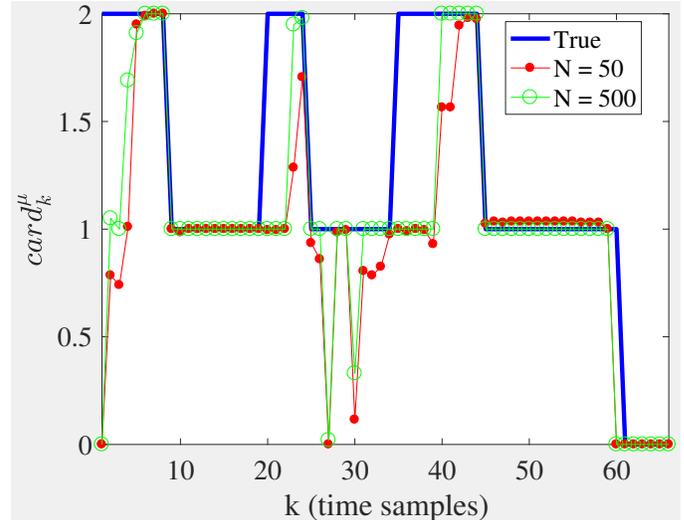
Fig. 3: **Mean cardinality**. The figure shows the true (blue thick line) and mean estimated cardinality using $N = 500$ (green circle line) and $N = 50$ (red dot line). It can be observed that although dense clutter is present, no false alarms are observed.

References

[1] W. Aftab, A. De Freitas, M. Arvaneh, and L. Mihaylova, "A Gaussian process approach for extended object tracking with random shapes and for dealing with intractable likelihoods," in *the Proceedings of the 22nd International Conference on Digital Signal Processing (DSP)*. IEEE, 2017, pp. 1–5.

[2] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stübler, M. Bach, P. Hatzelmann *et al.*, "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 666–673.

[3] F. Faion, M. Baum, and U. D. Hanebeck, "Tracking 3D shapes in noisy point clouds with random hypersurface models," in *the Proceedings of the 15th International Conference on Information Fusion*. IEEE, 2012, pp. 2230–2235.

[4] F. Septier, A. Carmi, and S. Godsill, "Tracking of multiple contaminant clouds," in *the Proceedings of the 12th International Conference on Information Fusion*. IEEE, 2009, pp. 1280–1287.

[5] B. Samuel and P. Robert, "Design and analysis of modern tracking systems," *London: Artech House*, 1999.

[6] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing*, vol. 25, pp. 1–16, 2014.

[7] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *arXiv preprint arXiv:1604.00970*, 2016.

[8] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems (Book)," *Norwood, MA: Artech House, 1999.*, 1999.

[9] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[10] K. Granström, S. Reuter, D. Meissner, and A. Scheel, "A multiple model PHD approach to tracking of cars under an assumed rectangular shape," in *the Proceedings of the 17th International Conference on Information Fusion*. IEEE, 2014, pp. 1–8.

[11] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino, "PHD extended target tracking using an incoherent X-band radar: Preliminary real-world experimental results," in *the Proceedings of the 17th International Conference on Information Fusion*. IEEE, 2014, pp. 1–8.

[12] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *the Proceedings of the IEEE International Symposium on Signal Processing and Information Technology*. IEEE, 2009, pp. 178–183.
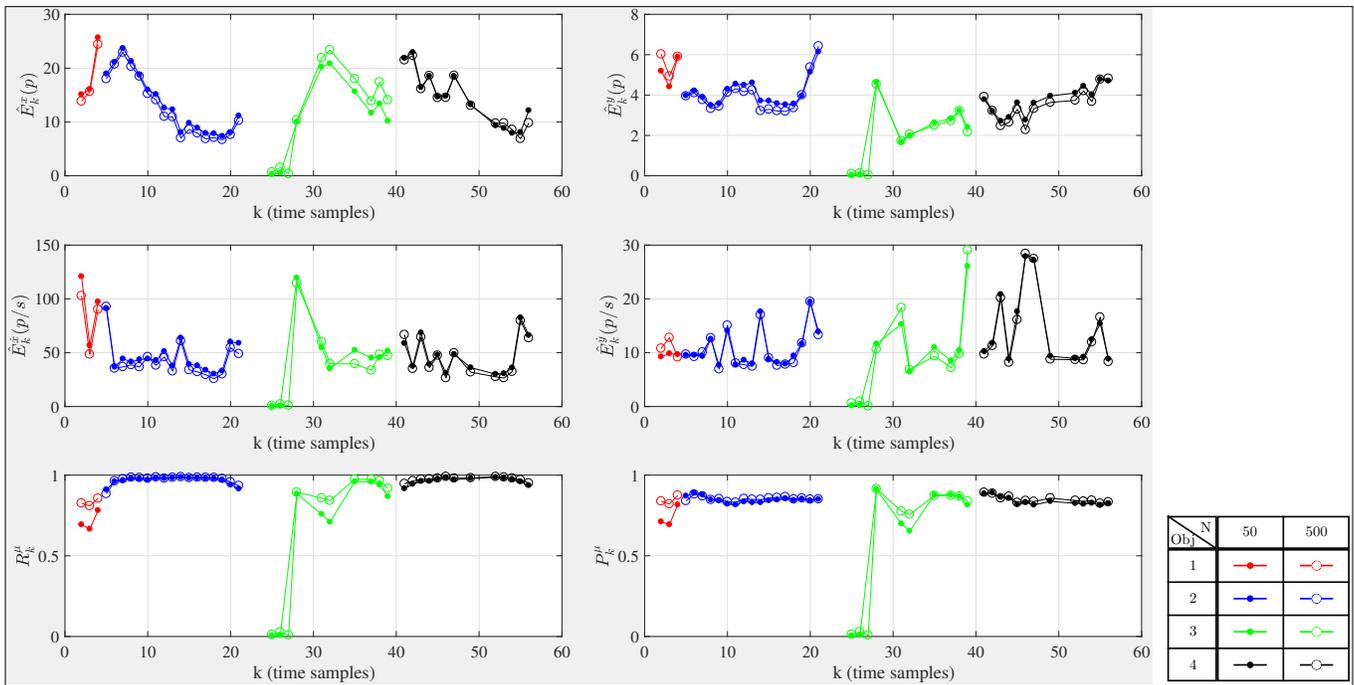
Fig. 4: **State Errors**. The state estimation errors are shown for the four moving objects (obj-1 in red, obj-2 in blue, obj-3 in green and obj-4 in black) for two different number of particles that is $N = 500$ (solid line with dots) and $N = 50$ (solid line with circles). The positional errors are given in the top two, the velocity errors in the middle and the shape errors are given in the bottom two subplots. The errors are depicted only for those time samples when both the ground truth and the estimated states are available. The distance unit is in the frame of reference of the image that is pixels (p).

[13] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, 2015.

[14] B. Lei, C. Li, and H. Ji, "Nonlinear maneuvering non-ellipsoidal extended object tracking using random matrix," in *the Proceedings of the 20th International Conference on Information Fusion*. IEEE, 2017, pp. 1–6.

[15] 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.

[16] K. Granström, S. Renter, M. Fatemi, and L. Svensson, "Pedestrian tracking using Velodyne data–Stochastic optimization for extended object tracking," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 39–46.

[17] M. Baum, B. Noack, and U. D. Hanebeck, "Mixture random hypersurface models for tracking multiple extended objects," in *the Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. IEEE, 2011, pp. 3166–3171.

[18] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer, "Multiple extended object tracking using Gaussian processes," in *the Proceedings of the 19th International Conference on Information Fusion (FUSION), 2016*. IEEE, 2016, pp. 868–875.

[19] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[20] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[21] D. Angelova, L. Mihaylova, N. Petrov, and A. Gning, "A convolution particle filtering approach for tracking elliptical extended objects," in *the Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 1542–1549.

[22] S. K. Pang, J. Li, and S. J. Godsill, "Detection and tracking of coordinated groups," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 472–502, 2011.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The Kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[24] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.