
Proceedings Paper:
Aftab, W., De Freitas, A., Arvaneh, M. et al. (1 more author) (2017) A Gaussian Process Approach for Extended Object Tracking with Random Shapes and for Dealing with Intractable Likelihoods. In: Proceedings of the 2017 International Conference on Digital Signal Processing (DSP). 2017 22nd International Conference on on Digital Signal Processing (DSP 2017), 23-25 Aug 2017, London, UK. IEEE .

10.1109/ICDSP.2017.8096087

A Gaussian Process Approach for Extended Object Tracking with Random Shapes and for Dealing with Intractable Likelihoods

Waqas Aftab, Allan De Freitas, Mahnaz Arvaneh and Lyudmila Mihaylova
Department of Automatic Control and System Engineering, The University of Sheffield, Sheffield, UK S13JD
Email:{ waftab1, ADeFreitas1, m.arvaneh, L.S.Mihaylova} @sheffield.ac.uk

***Abstract*—Tracking of arbitrarily shaped extended objects is a complex task due to the intractable analytical expression of measurement to object associations. The presence of sensor noise and clutter worsens the situation. Although a significant work has been done on the extended object tracking (EOT) problems, most of the developed methods are restricted by assumptions on the shape of the object such as stick, circle, or other axis-symmetric properties etc. This paper proposes a novel Gaussian process approach for tracking an extended object using a convolution particle filter (CPF). The new approach is shown to track irregularly shaped objects efficiently in presence of measurement noise and clutter. The mean recall and precision values for the shape, calculated by the proposed method on simulated data are around 0.9, respectively, by using 1000 particles.**

I. Introduction

Tracking extended objects aims to estimate the kinematic states and shape parameters of an object of interest using measurements reported by a sensor. Objects can be categorised based on the number of object measurements generated by the sensor per sample time i.e. when multiple or a single measurement is observed, the object is referred to as an extended object or a point object, respectively [1]. A large group of point objects moving in a coordinated fashion may also be modelled as an extended object [2].

The problem of tracking large groups is modelled in a similar way to extended object tracking (EOT) problems [2]. In [3] a Poisson likelihood model is used. In this case the measurement associations are resolved using Poisson models. In [4] the random matrix approach is used to model the extent as a random matrix. In [5] random finite sets methods are shown to track extended objects by modelling the target state and extent using finite set statistics (FISST). In all of the above mentioned approaches and mostly in general, the object shape is modelled using basic geometrical shapes e.g. stick [6] , circle [7], rectangle [8], ellipse [4], [9]. The tracking performance can be improved by considering a more detailed shape model, as in [10], [11] and the proposed approach.

The analytical form of likelihood cannot be derived for the EOT problem, as the measurements relate to the states non-linearly. The method proposed in this paper does not require explicit likelihood function for estimation. In [10] a Gaussian Process (GP) based extended Kalman filter (EKF) is proposed. The EKF tracks the object kinematics and states for both target contour and surface scenarios, however the performance degrades with increasing levels of non-linearity In [12], a Rao-Blackwellised particle filter (RBPF) based approach is used to sample the kinematics states of object and a GP regression based Kalman filter is used to track extent. This approach provides improved efficiency compared to [10], however in both the approaches the data association is not resolved. In [11], a star-convex random hypersurface model (RHM) is proposed to track star convex shapes. In surface models of GP-EKF, GP-RBPF and RHM, a distribution of measurements is required. The performance degrades when the statistical properties of actual measurements, which are not known in real world problems, are different from the modelled distribution. The data association is resolved in [13], where EKF is used in labelled multi-Bernoulli (LMB) framework and multiple extended objects are tracked. In the proposed approach highly non-linear target kinematics model is considered along with sensor clutter. The measurements are assumed to be coming from surface, however the statistical properties of the measurements are not required by the filter. In [8], [9] the convolution particle filter (CPF) is used to track an extended object but the object shape is assumed to be basic i.e. rectangular and ellipse. In the proposed approach the CPF is used to track a more complex shaped object.

In this paper a novel method for EOT is proposed. A CPF samples both the kinematics and extent states. As many GPs, as the number of particles, are trained on the extent samples. The GP is used to define the CPF kernel by estimating a hypersphere in measurement space. The CPF then resolves the data association in presence of sensor noise and clutter using the hypersphere and updates the particle weights to give an output estimate.

The rest of the paper is organized as follows. Section II describes the GP framework, Section III presents the convolution PF and Section IV formulates the problem using GP in CPF framework. The performance validation of the approach is done in Section V followed by conclusion in Section VI.

II. Gaussian Process

A GP is defined by a mean function and a covariance kernel. It is a stochastic process which maps an input to an output space. The covariance kernel is characterised by hyperparameters. An elaborate insight into different aspects of GPs and kernel design is given in [14]. Let the input and output spaces be represented by the random vectors $\boldsymbol{\theta}$ and $\boldsymbol{r}$, respectively. A GP $GP(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}'))$ is described by a

non-linear function $\boldsymbol{f}$

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{\theta}). \tag{1}$$

The GP learns the hyperparameters from a given set of training data, denoted as D. When trained on N input-output pairs, $D = \{(\boldsymbol{r}_1, \boldsymbol{\theta}_1), \cdots (\boldsymbol{r}_N, \boldsymbol{\theta}_N)\}$, the function values are normally distributed with the modelled GP mean and covariance i.e.

$$[\boldsymbol{f}(\boldsymbol{\theta}_1)^T \boldsymbol{f}(\boldsymbol{\theta}_2)^T \cdots \boldsymbol{f}(\boldsymbol{\theta}_N)^T]^T = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{2}$$

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbb{E}[\boldsymbol{f}(\boldsymbol{\theta})], \tag{3}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}} = \mathbb{E}[(\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{\mu}(\boldsymbol{\theta}))(\boldsymbol{f}(\boldsymbol{\theta}') - \boldsymbol{\mu}(\boldsymbol{\theta}'))^T], \tag{4}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \begin{bmatrix} k(\theta_1, \theta_1) & k(\theta_1, \theta_2) & ... & k(\theta_1, \theta_N) \\ k(\theta_2, \theta_1) & k(\theta_2, \theta_2) & ... & k(\theta_2, \theta_N) \\ \vdots & \vdots & \vdots & \vdots \\ k(\theta_N, \theta_1) & k(\theta_N, \theta_2) & ... & k(\theta_N, \theta_N) \end{bmatrix}, \tag{5}$$

where $\mathbb{E}[.]$ is the mathematical expectation operation. For each matrix element the first parameter of the kernel k is from $\boldsymbol{\theta}$ and the second parameter is from $\boldsymbol{\theta}'$. Each element of the matrix is found by evaluating the values using a kernel. The parameters of the kernel are called hyperparameters e.g. the hyperparameters of the kernel given in (16) are σ_f^2 and σ_a^2. To determine the optimum value of hyperparameters for the functional mapping, the likelihood $p(\boldsymbol{f}(\boldsymbol{\theta})|\boldsymbol{\theta}, \boldsymbol{D}) = \mathcal{N}(\boldsymbol{\mu}(\theta), \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}'} + \sigma^2 \boldsymbol{I}_N)$ is maximised over the hyperparameters. A trained GP, having learned the hyperparameters, can then predict the output vector at new locations given by $\boldsymbol{\theta}^\star$. Assume the new locations to be normally distributed as $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}^\star), \boldsymbol{C}(\boldsymbol{\theta}^\star, \boldsymbol{\theta}^{\star'}))$ and let the sensor measurement noise is $\sigma^2 \boldsymbol{I}_N$, where $\boldsymbol{I}_N$ is an N-dimensional identity matrix. The joint distribution of already given input locations and predictive locations is given by

$$\begin{bmatrix} \boldsymbol{f}(\boldsymbol{\theta}) \\ \boldsymbol{f}(\boldsymbol{\theta}^\star) \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}(\boldsymbol{\theta}) \\ \boldsymbol{\mu}(\boldsymbol{\theta}^\star) \end{bmatrix}, \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2 \boldsymbol{I}_N & \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}^\star} \\ \boldsymbol{C}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}} & \boldsymbol{C}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}^\star} \end{bmatrix} \Big). \tag{6}$$

Using (6) the predictive distribution can be written as

$$p(\boldsymbol{f}(\boldsymbol{\theta}^\star)|\boldsymbol{\theta}^\star, D) = \mathcal{N}(\boldsymbol{C}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}}(\boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{C}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}^\star} - \boldsymbol{C}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}}(\boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}^\star}). \tag{7}$$

III. Convolution Particle Filter

The CPF was first proposed in [15]. Consider the following model:

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k), \tag{8}$$

$$\boldsymbol{z}_k = \boldsymbol{g}_k(\boldsymbol{x}_{k-1}, \boldsymbol{\nu}_k), \tag{9}$$

where $\boldsymbol{x}_k$ is the state vector, $\boldsymbol{f}_k$ is the time update function for state with $\boldsymbol{w}_k$ process noise, $\boldsymbol{z}_k$ is the vector of measurements which is related to the state and the sensor error $\boldsymbol{\nu}_k$ through a function $\boldsymbol{g}_k$. The subscript k represents a discrete time step. The aim is to estimate the posterior density of the state given all measurements up to the current time step k as given below

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k})}{p(\boldsymbol{z}_{1:k})}. \tag{10}$$

A discrete set of points $\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{z}}_{1:k}$ can be simulated by updating the initial particle set $\tilde{\boldsymbol{x}}_0$ with the time update in (8) and by simulating measurements with (9). The empirical estimates from these point estimates are given below

$$\mu_k^N = \frac{1}{N} \sum_{n=1}^{N} \delta_{(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{z}}_{1:k})}, \nu_k^N = \frac{1}{N} \sum_{n=1}^{N} \delta_{\tilde{\boldsymbol{z}}_{1:k}}, \tag{11}$$

where $\delta_{(.)}$ is Dirac measure. The estimates of the posterior distribution can be obtained by the convolution of empirical measures with a kernel (K_h) where h is kernel bandwidth.

$$p^N(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) = K_h^{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}}_{1:k})} * \mu_k^N = \frac{1}{N} \sum_{n=1}^{N} K_h^{\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}^{(n)}) K_h^{\tilde{\boldsymbol{z}}}(\tilde{\boldsymbol{z}}_{1:k} - \tilde{\boldsymbol{z}}_{1:k}^{(n)}), \tag{12}$$

$$p^N(\boldsymbol{z}_{1:k}) = K_h^{\tilde{\boldsymbol{z}}_{1:k}} * \nu_k^N = \frac{1}{N} \sum_{n=1}^{N} K_h^{\tilde{\boldsymbol{z}}}(\tilde{\boldsymbol{z}}_{1:k} - \tilde{\boldsymbol{z}}_{1:k}^{(n)}), \tag{13}$$

$$K_h^{\tilde{\boldsymbol{z}}}(\tilde{\boldsymbol{z}}_{1:k} - \tilde{\boldsymbol{z}}_{1:k}^{(n)}) = \prod_{l=1}^{k} K_h^{z}(z_l - \tilde{z}_l^{(n)}), \tag{14}$$

where $K_h^{\tilde{\boldsymbol{x}}}$ and $K_h^{\tilde{\boldsymbol{z}}}$ are Parzen-Rozenblatt kernels of appropriate dimensions. The posterior, also called CPF density estimator, can be estimated using (10) and (12) - (14) as given below

$$p^N(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{\sum_{n=1}^{N} K_h^{\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}^{(n)}) K_h^{\tilde{\boldsymbol{z}}}(\tilde{\boldsymbol{z}}_{1:k} - \tilde{\boldsymbol{z}}_{1:k}^{(n)})}{\sum_{n=1}^{N} K_h^{\tilde{\boldsymbol{z}}}(\tilde{\boldsymbol{z}}_{1:k} - \tilde{\boldsymbol{z}}_{1:k}^{(n)})}. \tag{15}$$

IV. GP-CPF Approach

In order to solve the EOT problem, the centre of the object (COO) is tracked simultaneously with its extent, where the extent is modelled as a function of angles respect to COO. Modelling in this way converts the complex problem of associating the measurements to the respective point objects to a relatively simple problem.

A. Covariance Kernel

The covariance kernel is the core of GP based predictive models. The kernel is designed to predict radial values at input angles other than $\boldsymbol{\theta}^b$. The kernel is required to be periodic in the θ domain. The predicted values are required to be smooth between the two neighbouring angles, and uncorrelated with all other angles, in order to allow the output shape to be irregular. This can be achieved by determining pre-hand a set of hyperparameters that allow this behaviour. As a result there is no need to optimise the hyperparameters online and the processing complexity is reduced. A periodic kernel, inspired from the Von Mises distribution, is designed

$$k_{vm}(\theta_i, \theta_j) = \sigma_f^2 e^{\sigma_a^2 cos(\theta_i - \theta_j)}, \tag{16}$$

where σ_f^2, σ_a^2 control the variance of the kernel amplitude.

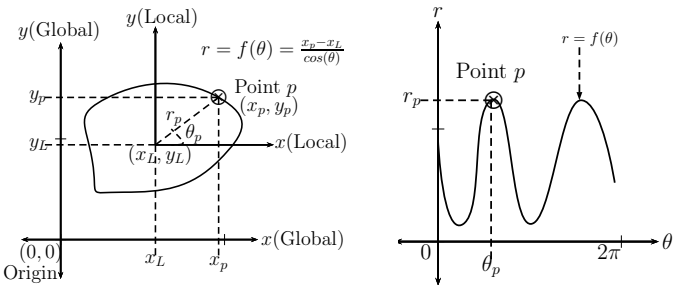
B. Crowd Extent Model

The shape of the crowd is assumed to be star convex and is modelled using a GP similar to [10]. The crowd extent is modelled as a function of angle from COO i.e.

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{\theta}), \tag{17}$$

where $\boldsymbol{\theta}$ can vary from 0 to 2π. The extended object can take any arbitrary shape as given in Fig. 1a. Two coordinate frames, namely global and local, along with their relationship are also shown in Fig. 1a. The radial function is modelled in local frame. The function $\boldsymbol{f}$ can be visualized by looking at Fig. 1b. The GP is used to map this function. The measurement vector $\boldsymbol{z}_k$ observed by sensor at time k for $l = 1, 2, ..., L$ measurements can be written as

$$\boldsymbol{z}_k = [x_{k,1}^{\boldsymbol{z}}, y_{k,1}^{\boldsymbol{z}}, x_{k,2}^{\boldsymbol{z}}, y_{k,2}^{\boldsymbol{z}}, \cdots, x_{k,L}^{\boldsymbol{z}}, y_{k,L}^{\boldsymbol{z}}]^T, \tag{18}$$



(a) Crowd Extent Model (b) Radial Function

Fig. 1: (a) Two coordinate systems are depicted namely Global and Local. Sensor measurements and filter output are given in the Global whereas the GP is modelled in Local coordinates. The origin of Local is at the centre of the extended object. (b) This figure visualizes the radial function r of Fig 1a (Not scaled). The point P is shown in cartesian frame in Fig. 1a.

where $x^{\boldsymbol{z}}_{k,l}, y^{\boldsymbol{z}}_{k,l}$ are cartesian coordinates of the i^{th} measurement. Let $(r^{\boldsymbol{z}}_{k,l}, \theta^{\boldsymbol{z}}_{k,l})$ be respective polar coordinates and (x^c_k, y^c_k) be origin of local frame then,

$$x^{\boldsymbol{z}}_{k,l} = x^c_k + r^{\boldsymbol{z}}_{k,l} cos(\theta^{\boldsymbol{z}}_{k,l}) + \nu^x_k, \quad \nu^x_k \sim \mathcal{N}(0, \sigma^2_x), \quad (19)$$

$$y^{\boldsymbol{z}}_{k,l} = y^c_k + r^{\boldsymbol{z}}_{k,l} sin(\theta^{\boldsymbol{z}}_{k,l}) + \nu^y_k, \quad \nu^y_k \sim \mathcal{N}(0, \sigma^2_y). \quad (20)$$

Substituting (17) in (19,20)

$$x^z_{k,l} = x^c_k + f(\theta^z_{k,l}) cos(\theta^z_{k,l}) + \nu^x_k, \quad (21)$$

$$y^z_{k,l} = y^c_k + f(\theta^z_{k,l}) sin(\theta^z_{k,l}) + \nu^y_k. \quad (22)$$

The function $\boldsymbol{f}$ is modelled as a GP i.e. $\boldsymbol{f}(\boldsymbol{\theta}) \sim GP(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}'))$. The mean is modelled to be unknown and constant, $\boldsymbol{\mu}(\boldsymbol{\theta}) = r$. As given in [10], the unknown constant mean is included in the model by setting prior on the unknown mean $r = \mathcal{N}(0, \sigma^2_r)$. The GP model is then zero mean i.e. $\boldsymbol{f}(\boldsymbol{\theta}) \sim GP(0, \boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}'))$. The covariance kernel (16) is modified as given below.

$$k(\theta_i, \theta_j) = k_{vm}(\theta_i, \theta_j) + \sigma^2_r \quad (23)$$

Use (23) and (5) to determine $\boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\theta}')$.

C. State Vector

The states inferred at each time step k are assumed to be in $2D$ and are represented by state vector $\boldsymbol{x}_k$:

$$\boldsymbol{x}_k = [\boldsymbol{x}^K_k, \boldsymbol{x}^E_k]^T, \quad (24)$$

where $\boldsymbol{x}^K_k$ represent kinematics of COO and $\boldsymbol{x}^E_k$ denotes extent states at B input locations,

$$\boldsymbol{x}^K_k = [x^c_k, \dot{x}^c_k, y^c_k, \dot{y}^c_k]^T, \boldsymbol{x}^E_k = [r^1_k, r^2_k, ..., r^B_k]^T. \quad (25)$$

where x^c_k, y^c_k and $\dot{x}^c_k, \dot{y}^c_k$ are position and velocity of COO. r^i_k are the radial values of the object corresponding to input vector $\boldsymbol{\theta}^b$

$$\boldsymbol{\theta}^b = [\theta^1, \theta^2, \cdots, \theta^B]^T, \quad \theta^i = (i-1)\frac{2\pi}{B}. \quad (26)$$

D. State Sampling

The kinematics of the crowd centre is modelled as a correlated velocity model [8] using the following time update;

$$\boldsymbol{x}^K_k = \boldsymbol{F}^K \boldsymbol{x}^K_{k-1} + \boldsymbol{w}^K_k, \quad (27)$$

where $\boldsymbol{w}^K_k$ is the model process noise, $(.)^K$ represents that the vector / matrix corresponds to evolution of the kinematic state of the crowd. The state transition matrix $\boldsymbol{F}^K$ is

$$\boldsymbol{F}^K = \begin{bmatrix} 1 & \frac{1}{\alpha}(1 - e^{-\alpha \Delta T}) & 0 & 0 \\ 0 & e^{-\alpha \Delta T} & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{\alpha}(1 - e^{-\alpha \Delta T}) \\ 0 & 0 & 0 & e^{-\alpha \Delta T} \end{bmatrix}, \quad (28)$$

where α is the correlation constant for velocity. The system process noise is defined as,

$$\boldsymbol{Q}^K_k = 2\alpha \begin{bmatrix} \sigma^2_{v_x} q_{11} & \sigma^2_{v_x} q_{12} & 0 & 0 \\ \sigma^2_{v_x} q_{21} & \sigma^2_{v_x} q_{22} & 0 & 0 \\ 0 & 0 & \sigma^2_{v_y} q_{11} & \sigma^2_{v_y} q_{12} \\ 0 & 0 & \sigma^2_{v_y} q_{21} & \sigma^2_{v_y} q_{22} \end{bmatrix}, \quad (29)$$

where $\sigma^2_{v_x}$ and $\sigma^2_{v_y}$ are variances of the crowd centre velocities in respective coordinates and;

$$q_{11} = \frac{1}{2\alpha^3}(4e^{-\alpha \Delta T} - 3 - e^{-2\alpha \Delta T} + 2\alpha \Delta T), \quad (30)$$

$$q_{12} = q_{21} = \frac{1}{2\alpha^2}(e^{-2\alpha \Delta T} + 1 - 2e^{-\alpha \Delta T}), \quad (31)$$

$$q_{22} = \frac{1}{2\alpha}(1 - e^{-2\alpha \Delta T}). \quad (32)$$

The state sampling equation for crowd extent is performed as a random walk around the state at the previous scan as

$$\boldsymbol{x}^E_k = \boldsymbol{x}^E_{k-1} + \boldsymbol{w}^E_k, \quad \boldsymbol{w}^E_k \sim \mathcal{N}(0, \boldsymbol{Q}^E_k = \sigma^2_e \boldsymbol{I}_{B \times B}). \quad (33)$$

The $\boldsymbol{Q}^E_k$ is a diagonal matrix with variance σ^2_e at the main diagonal. The object can have any shape, which in other words means that all the radial values can be uncorrelated. Smaller variance means the filter takes more time to lock the extent parameters. The number of particles for filter convergence will increase if variance is large.

E. Measurement Simulation and Weight Update

For each sampled state, simulate the measurements as

$$\boldsymbol{\Gamma}^n_k = \boldsymbol{I}_{B \times B} \otimes ([x^c_k, y^c_k]^T)^n + \boldsymbol{J}^n_k (\boldsymbol{x}^E_k)^n + \boldsymbol{\nu}_k, \quad (34)$$

$$\boldsymbol{\nu}_k \sim \mathcal{N}\Big(0, \boldsymbol{I}_{B \times B} \otimes \begin{bmatrix} \sigma^2_x & 0 \\ 0 & \sigma^2_y \end{bmatrix}\Big)$$

$$\boldsymbol{J}^n_k = \boldsymbol{I}_{B \times B} \otimes \begin{bmatrix} cos(\theta^b) & sin(\theta^b) \end{bmatrix}^T, \theta^b \in \boldsymbol{\theta}^b \quad (35)$$

where $\otimes$ denotes kronecker product, $\boldsymbol{\Gamma}^n_k$ represents a hypersphere in measurement space for n^{th} particle at time k. All measurements within this hypersphere are considered gated with the n^{th} particle. This hypersphere lies in 2D so it can be referred as a polygon P^n_k. The GP associated with each particle is trained on $\boldsymbol{\Gamma}^n_k$. Let $\boldsymbol{r}^n_{\boldsymbol{\Gamma}_k}$ represent that radial values of n^{th} particle with respect to center $([x^c_k, y^c_k]^T)^n$, $\boldsymbol{y}^n_k$ represent the radial values of sampled measurement polygon (P^n_k) at time k for particle n. Then using (7) we can write

$$[\boldsymbol{\theta}^n_{\boldsymbol{\Gamma}_k}, \boldsymbol{r}^n_{\boldsymbol{\Gamma}_k}] = cart2pol(\boldsymbol{\Gamma}^n_k - \boldsymbol{I}_{B \times B} \otimes ([x^c_k, y^c_k]^T)^n) \quad (36)$$

$$\boldsymbol{y}^n_k = \boldsymbol{C}_{\boldsymbol{\theta}^{\boldsymbol{y}} \boldsymbol{\theta}^b} (\boldsymbol{C}_{\boldsymbol{\theta}^b \boldsymbol{\theta}^b})^{-1} \boldsymbol{r}^n_{\boldsymbol{\Gamma}_k}. \quad (37)$$

where $\boldsymbol{\theta}^{\boldsymbol{y}} = [\theta^1, \theta^2, \cdots, \theta^Y]^T$ represents the input angles of sampled polygon P^n_k and $cart2pol$ is a function which converts coordinates from cartesian to polar frame. The size of $\boldsymbol{\theta}^{\boldsymbol{y}}$ effects the CPF kernel performance and is modelled large compared to the $\boldsymbol{\theta}^b$. For gating the measurements with n^{th} particle, find bearing $\boldsymbol{\theta}^n_{k^L}$ of all the measurements with respect to the particle n. The bearing calculation for l^{th} measurement with n^{th} particle is given below

$$[\theta^n_{k^l}, r^n_{k^l}]^T = cart2pol([x^{\boldsymbol{z}}_{k,l}, y^{\boldsymbol{z}}_{k,l}]^T - ([x^c_k, y^c_k]^T)^n). \quad (38)$$

The radial extent of n^{th} particle at these measurements is

$$\boldsymbol{r}^n_{\boldsymbol{z}_k} = \boldsymbol{C}^n_{\boldsymbol{\theta}^L_k \boldsymbol{\theta}^b}(\boldsymbol{C}_{\boldsymbol{\theta}^b\boldsymbol{\theta}^b})^{-1}\boldsymbol{y}^n_k. \quad (39)$$

The measurements with radial values less than $\boldsymbol{r}^n_{\boldsymbol{z}_k}$ are gated with that particle. Define a uniform kernel with interval support at P^n_k or λ for observation and clutter, respectively.

$$K^{\boldsymbol{y}^n_k}_h(\boldsymbol{z}) = \begin{cases} \mathcal{U}_{P^n_k}(\boldsymbol{z}), & \text{if } \boldsymbol{z} \in P^n_k \\ \mathcal{U}_\lambda(\boldsymbol{z}), & \text{otherwise} \end{cases} \quad (40)$$

where $\mathcal{U}_{P^n_k}(\boldsymbol{z})$ is a uniform kernel with support defined over polygon P^n_k and $\mathcal{U}_\lambda(\boldsymbol{z})$ is a uniform kernel with support defined over complete surveillance area and it represents clutter measurements. Also

$$\mathcal{U}_{P^n_k}(\boldsymbol{z}) = \frac{1}{\text{Polygon } P^n_k \text{ Area}}, \mathcal{U}_\lambda(\boldsymbol{z}) = \frac{1}{\text{Surveillance Area}}. \quad (41)$$

The kernel function returns different values for different particles based on the area of polygon i.e. adaptive CPF [8]. Using (18), the weight update at scan k for a given particle n is

$$w^{(n)}_k = w^{(n)}_{k-1} \prod_{l=1}^{L} K^{\boldsymbol{y}^n_k}_h(\boldsymbol{z}^l_k). \quad (42)$$

F. Estimation

The conditional state density for CPF can be written as:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k})}{\int p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) d\boldsymbol{x}_k} \quad (43)$$

Along the lines of adaptive CPF modelled in [9], the kinematic and extent states are sampled separately. The estimate equation is given below;

$$p^N_k(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{\sum_{i=1}^N K^{\boldsymbol{x}}_h(\boldsymbol{x}_k - \boldsymbol{x}^i_k) K^{\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{k}}}_h(\boldsymbol{z}_{1:k})}{\sum_{i=1}^N K^{\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{k}}}_h(\boldsymbol{z}_{1:k})}, \quad (44)$$

$$K^{\boldsymbol{x}}_h(\boldsymbol{x}_k - \boldsymbol{x}^i_k) = K^{\boldsymbol{x}^K}_h(\boldsymbol{x}^K_k - (\boldsymbol{x}^K_k)^i) K^{\boldsymbol{x}^E}_h(\boldsymbol{x}^E_k - (\boldsymbol{x}^E_k)^i), \quad (45)$$

$$K^{\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{k}}}_h(\boldsymbol{z}_{1:k}) = \prod_{j=1}^{K} K^{\boldsymbol{y}^n_k}_h(\boldsymbol{z}_j), \quad (46)$$

where $K^{\boldsymbol{x}^K}_h$, $K^{\boldsymbol{x}^E}_h$ and $K^{\boldsymbol{y}^n_k}_h$ are Parzen-Rosenblatt kernels. The state estimate is given below

$$\hat{\boldsymbol{x}}^K_k = \frac{\sum_{i=1}^N w^{(i)}_k (\boldsymbol{x}^K_k)^{(i)}}{\sum_{i=1}^N w^{(i)}_k}, \hat{\boldsymbol{x}}^E_k = \frac{\sum_{i=1}^N w^{(i)}_k (\boldsymbol{x}^E_k)^{(i)}}{\sum_{i=1}^N w^{(i)}_k}. \quad (47)$$

Choose number of output extent states, B_o, greater than number of basis , B. Use (39) to predict the output extent estimate $\hat{\boldsymbol{x}}^E_{k,o}$ for B_o basis. The output state is then $\hat{\boldsymbol{x}}_{k,o} = [\hat{\boldsymbol{x}}^K_k, \hat{\boldsymbol{x}}^E_{k,o}]^T$. The GP-CPF recursion is summarized in Table I.

V. Performance Validation

Simulation. The simulations are performed on large group of point objects inside an irregular pentagon. The mean number of measurement sources are Poisson distributed with mean $\lambda = 420$, the total number of scans $K = 100$ and sampling time is $\Delta T = 0.125s$. The initial state for centre of motion is $[100m, 0m/s, 50m, 0m/s]^T$. The velocity correlation time constant and standard deviation are $T_{cv} = 15s$, $\sigma_{v,x} = \sigma_{v,y} = 10m/s$ and extent radial dynamics parameters are $\sigma_r = 0.1m$ per time step. The sensor measurement error is $\sigma_x = \sigma_y = 0.1m$. The clutter density is $\rho = 1 \times 10^{-3}$ within a circular surveillance region of radius $100m$.

GP-CPF Parameters. The kinematic parameters are matched to simulation. The initial states are normally distributed around initial state $\boldsymbol{x}^K_0 = \mathcal{N}([x^c_2, \dot{x}_0, y^c_2, \dot{y}_0]^T, [.25m, 1m/s, .25m, 1m/s]^T)$ and $\boldsymbol{x}^E_0 = \mathcal{N}([R_2, \cdots R_2]^T, [1e^{-6}m, \cdots 1e^{-6}m]^T)$ where (x^c_i, y^c_i) are coordinates of COO at $k = i$, $\dot{x}_0 = \frac{x^c_2 - x^c_1}{\Delta T}, \dot{y}_0 = \frac{y^c_2 - y^c_1}{\Delta T}$ and R_2 is maximum radial value at scan 2. All the particles are initialized to equal weights. The COO is determined using k-means clustering during initialization i.e. when $k = 1, 2$. The extent process noise standard deviation is $\sigma^2_e = \frac{10}{3}$, hyperparameters values are $\sigma^2_a = \frac{1}{40}$, $\sigma^2_r = 1$ and $\sigma^2_f = 30$. The number of particles is $N = 1000$, number of basis is $B = 16$, number of points of polygon P^n_k is $Y = 1440$, and number of output basis is $B_o = 1440$.

TABLE I: GP-CPF Recursion

1	for $k \leq 2$, $\boldsymbol{\theta}_B = 0 : 360/B : 360$ find $\boldsymbol{x}_0$ as given in subsection V
2	for $k = 3$ find $\tilde{\boldsymbol{x}}^n_0 = \mathcal{N}(\boldsymbol{x}_0, \sigma^2_{\boldsymbol{x}_0})$, $w^n_0 = \frac{1}{N}$
3	for $k > 3$ Re-sample : Residual Re-sampling as in [16].
4	for $k \geq 3$
4a	State Sample: for $n = 1, 2, ...N$ determine $(\tilde{\boldsymbol{x}}^K_k)^n \sim f^K_k((\tilde{\boldsymbol{x}}^K_{k-1})^n, (\boldsymbol{w}^K_{k-1})^n)$ $(\tilde{\boldsymbol{x}}^E_k)^n \sim f^E_k((\tilde{\boldsymbol{x}}^E_{k-1})^n, (\boldsymbol{w}^E_{k-1})^n)$
4b	Measurement Simulation : Add measurement noise using (34)
4c	Measurement Gating : Find $\boldsymbol{\theta}^n_{k^l}$ and $\boldsymbol{r}^n_{k^l}$ using (38). Use (39) to find $\boldsymbol{r}^n_{P_k}$. Gate = success, if $\boldsymbol{r}^n_{P_k} \geq \boldsymbol{r}^n_{k^l}$
4d	Weight Update : for $n = 1, 2, ...N$ use (40) to find $w^n_k = \prod_{l=1}^L K^{\boldsymbol{y}^n_k}_h(z_{k,l}) \times w^n_{k-1}$
4e	Normalize Weight : for $n = 1, 2, ...N$ determine $w^n_k = \frac{w^n_k}{\sum_{n=1}^N w^n_k}$
4f	Estimation : $p^n_k(\boldsymbol{x}_k|\boldsymbol{z}_{1:K}) = \sum_{n=1}^N w^n_k K^{\boldsymbol{x}}_h(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}^n_k)$
4g	Output : For $\boldsymbol{\theta}_o = 0 : 360/B_o : 360$, use (39) to predict $\hat{\boldsymbol{x}}^E_{k,o}$. Output estimate is $\hat{\boldsymbol{x}}_{k,o} = [\hat{\boldsymbol{x}}^K_k, \hat{\boldsymbol{x}}^E_{k,o}]^T$

Results. The results are compiled for 50 Monte Carlo runs. The evaluation of COO estimates is done using root mean square errors (RMSE) over a number of Monte Carlo simulation runs (N_{MC})

$$RMSE_q = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \sum_{k=1}^{K} (q_k - \hat{q}_k)^2}, \quad (48)$$

where q_k represents the ground truth for each of the four kinematic states at time k and $\hat{q}_k$ corresponds to their respective estimates. The mean Precision (P_μ) and Recall (R_μ) graphs are produced for evaluating the shape estimates. This scheme has been used in computer vision for evaluating rectangular objects detection performance [17]. The R_μ describes the ground truth area that has been correctly recalled by the algorithm whereas the P_μ illustrates incorrectly detected area. If E represents the estimated shape and T represents ground truth then the formulae for N_{MC} Monte Carlo runs are given below

$$R_\mu = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \sum_{k=1}^{K} \frac{Area(T_k \cap E_k)}{Area(T_k)}, \quad (49)$$

$$P_\mu = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \sum_{k=1}^{K} \frac{Area(T_k \cap E_k)}{Area(E_k)}. \quad (50)$$

The results from the above simulation are shown in Fig. 2. The CPF kernel is uniform and it weighs the particles based on overlap with measurements. Hence sometimes the COO estimates are slightly away from true COO whereas the shape

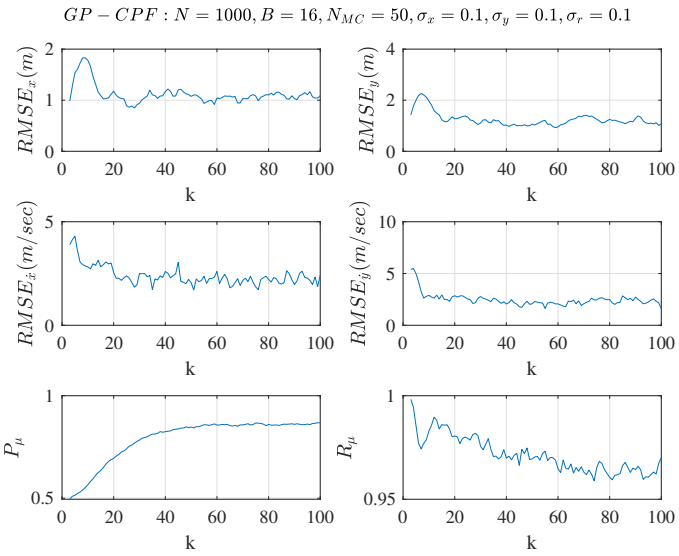


Fig. 2: Results of 50 Monte Carlo Runs. The average positional RMSE is around $1m$ and velocity RMSE is around $2m/sec$. The P_μ is around 0.87 most of the time which means almost 13% of the estimated shape is different from the ground truth. The R_μ is around 0.96 which means that almost 96% of ground truth shape has been recalled.

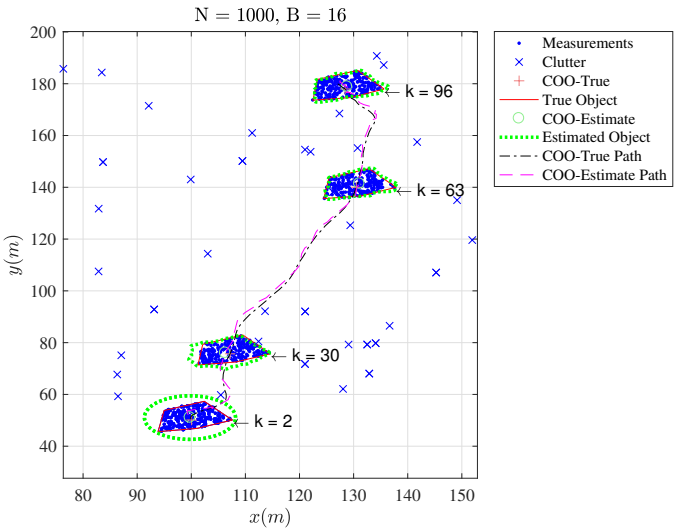


Fig. 3: Snapshots of tracking at $k = 2, 30, 63$ and 96. The true and estimated objects and paths are shown in the figure. It can be observed that the shape estimates improve progressively from an initial circular to ground truth shape.

estimate remains good at the same time. The figure 3 shows snapshots of simulated and estimated objects at time steps $k = 2, 30, 63$ and 96. The estimate of the path and the shape of extended object is close to ground truth in all steps given that the shape was initialized as a circle (at $k = 2$). To increase comprehensiveness of the figure the shapes have been displayed at chosen time steps. The filter runs in real-time and the mean simulation time is $49s$ for 100 time steps. The program was run on MATLAB R2016b on a Windows 10 (64 bit) Desktop computer installed with an intel(R) Core(TM) i5-6500 CPU @ 3.20GHz(4 CPUs) and 8GB RAM.

VI. Conclusions

This paper proposes a GP based approach in a CPF framework to track an extended object with arbitrary shape using data from a noisy sensor with clutter. The GP keeps track of the object extent using state samples and measurement simulations of CPF. As a result the filter is able to work in real time when the extended object is described by a highly non-linear kinematic model with a positional accuracy of around $1m$. The shape precision and recall estimates are around 0.9. Future work will be focused on real-time tracking of multiple extended objects.

Acknowledgements. We acknowledge the support from the Pakistan Air Force (Govt. of Pakistan), the Dept. of ACSE (University of Sheffield) and the EC Seventh Framework Programme [FP7 20132017] TRAcking in compleX sensor systems (TRAX) Grant agreement no.: 607400.

References

[1] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications (to appear)," *ISIF Journal of Advances in Information Fusion*, 2017.

[2] 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.

[3] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *Proceedings of the Optics & Photonics 2005*, 2005, pp. 59 130R–59 130R.

[4] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, 2011.

[5] R. Mahler, "Random set theory for target tracking and identification," in *Multisensor Data Fusion*. CRC press, 2001.

[6] M. Baum, F. Faion, and U. D. Hanebeck, "Modeling the target extent with multiplicative noise," in *Proceedings of the 15th International Conference on Information Fusion*, 2012, pp. 2406–2412.

[7] M. Baum, V. Klumpp, and U. D. Hanebeck, "A novel Bayesian method for fitting a circle to noisy points," in *Proceedings of the 13th Conference on Information Fusion*, 2010, pp. 1–6.

[8] A. De Freitas, L. Mihaylova, A. Gning, D. Angelova, and V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering," *Automatica*, vol. 69, pp. 380–394, 2016.

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

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

[11] M. Baum and U. D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex rhms," in *Proceedings of the 14th International Conference on Information Fusion*, 2011, pp. 1–8.

[12] E. Özkan, N. Wahlström, and S. J. Godsill, "Rao-blackwellised particle filter for star-convex extended target tracking models," in *Proceedings of the 19th Conference on Information Fusion*, 2016, pp. 1193–1199.

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

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

[15] V. Rossi and J.-P. Vila, "Nonlinear filtering in discrete time: A particle convolution approach," *Annales de lInstitut de Statistique de lUniversité de Paris*, vol. 50, no. 3, pp. 71–102, 2006.

[16] 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.

[17] 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.