

Spider monkey optimisation assisted particle filter for robust object tracking

ISSN 1751-9632
 Received on 4th July 2016
 Revised 21st October 2016
 Accepted on 15th November 2016
 E-First on 7th March 2017
 doi: 10.1049/iet-cvi.2016.0201
www.ietdl.org

Rajesh Rohilla¹ , Vanshaj Sikri¹, Rajiv Kapoor¹

¹Department of Electronics and Communication, Delhi Technological University, Bawana Road, Delhi 110042, India

 E-mail: rajesh@dce.ac.in

Abstract: Particle filters (PFs) are sequential Monte Carlo methods that use particle representation of state-space model to implement the recursive Bayesian filter for non-linear and non-Gaussian systems. Owing to this property, PFs have been extensively used for object tracking in recent years. Although PFs provide a robust object tracking framework, they suffer from shortcomings. Particle degeneracy and particle impoverishment brought by the resampling step result in abysmal construction of posterior probability density function (PDF) of the state. To overcome these problems, this work amalgamates two characteristics of population-based heuristic optimisation algorithms: exploration and exploitation with PF implementing dynamic resampling method. The aim of optimisation is to distribute particles in high-likelihood area according to the cognitive effect and improve quality of particles, while the objective of dynamic resampling is to maintain diversity in the particle set. This work uses very efficient spider monkey optimisation to achieve this. Furthermore, to test the efficiency of the proposed algorithm, experiments were carried out on one-dimensional state estimation problem, bearing only tracking problem, standard videos and synthesised videos. Metrics obtained show that the proposed algorithm outplays simple PF, particle swarm optimisation based PF, and cuckoo search based PF, and effectively handles different challenges inherent in object tracking.

1 Introduction

Object tracking is one of the major research areas in computer vision. The idea is to first detect one or more targets, and then obtain a record of trajectory of those targets in the video sequence. Over the years, object tracking has created a niche for itself in computer vision applications ranging from vehicle tracking [1] and monitoring and surveillance systems [2], to drive assistance [3] and human tracking [4].

However, the task of object tracking is suffused with challenges. Often little information is made available about the target. Even if the target information is given, the target appearance may change over time and space. Occlusion, background clutter, camouflaged foreground, and changing illumination in frame mar the efforts to achieve robust tracking. Furthermore, varying scale, rotating object, moving camera, and shadowing make the already difficult task more arduous.

To efficiently tackle the difficulties arising in vision tracking, researchers have proposed algorithms that can be roughly categorised under deterministic approach or stochastic approach. Deterministic methods localise tracked object in each frame by iteratively searching for region which maximises the similarity between the tracked region and target. Though these methods are computationally efficient, they may converge to a local optima. Additionally, they are prone to background clutter and occlusion. Lately, [5, 6] have been proposed for tracking objects. These techniques use online learning mechanisms to track target. In [6], algorithm proposed divides the task of object tracking into three components: tracking, learning, and detection. A novel learning method has been proposed called P–N learning that estimates the errors in detection by a pair of experts named P-expert (for estimating missed target detections) and N-expert (for estimating false alarms).

However, this research focusses mainly on the drawbacks of particle filters (PFs) (a stochastic approach) that limit their efficiency in tracking objects and how nature-inspired optimisation techniques can overcome those drawbacks. PF was first reported in [7] as bootstrap filter. Over the years improved versions of PFs have been introduced as condensation filter [8], interacting particle approximation [9], and sequential importance sampling [10].

PF represents the posterior PDF of system's state by random particles with associated weights and estimates the final state based on these particles. Keeping this in mind, PF views the tracking algorithm as a state estimation problem under Bayesian model and is considered to be one of the most useful tools to estimate Bayesian models with non-linear and non-Gaussian noise [10, 11].

Nevertheless, increased efficiency of PF entails increased computation cost that can create a delay in applications involving real-time systems. The increased dimensionality of state space adds to the computational burden associated with PFs.

Apart from these challenges, particle degeneration is a common problem that limits the efficiency of PFs. Degeneracy is a phenomena in which after few iterations most of the particles have negligible weight and their contribution towards the state approximation is almost zero. Degeneracy causes wastage of computational efforts invested in updating particles having negligible weights. Two common methods to deal with degeneracy are:

- a. Resampling.
- b. Good choice of importance density.

Resampling is a method by which particles with small weights are eliminated from the state vector estimate and are replaced with particles having large weights. The authors in [12–15] present systematic resampling, adaptive resampling, residual resampling, multinomial resampling, and stratified resampling methodologies, respectively. Nonetheless, these resampling methods partially solve the particle degeneracy problem. Instead, resampling creates a new problem of sample impoverishment. Resampling strategies lead to decreased diversity in the particle set. This results in poor construction of posterior PDF. Lately, more efficient approaches for resampling particles have been advanced by the researchers. Zuo [16] discusses a dynamic resampling (DR) method in which resampling operation is only performed on part of the particles in a step-by-step manner and the number of particles to be resampled is decided dynamically by a termination criterion based on the effective sample size. New born particles produced by resampling operation are helpful in alleviating degeneracy, whereas particles that are not resampled maintain the diversity. This technique has

been used in this work to resample particles in spider monkey optimisation based PF (SMO-PF). In [17], authors have introduced a novel resampling technique named as spline resampling that consists of two parts: spline transformation of weights and the spread transformation of states. The former is based on a spline transformation and achieves accurate particle filtering and latter is based on point spread transformation on states of particles to prevent sample impoverishment. Zhai and Yeary [18] proposes a PF that uses Markov Chain Monte Carlo method based Metropolis–Hastings algorithm to resample particles in the PF.

To address the issue of sample impoverishment, researchers have also developed more accurate posterior distribution. In [19], authors have developed unscented PF that is a parametric/non-parametric hybrid of unscented Kalman filter (UKF) and PF so that the PF provides the general probabilistic framework to handle non-linearity and non-Gaussianity and UKF generates better proposal distribution.

Lately, nature-inspired optimisation algorithms have captured great deal of attention of the researchers working on PF-based object tracking. These algorithms provide strategies that guide the search process in the fitness space. The given paper proposes an object tracking algorithm based on PF and SMO algorithm that models the foraging behaviour of spider monkeys (SMs). The proposed work parallels the line of thought used in [20] and optimises the distribution of sample particles before the DR step. Extensive experiments were carried out to test the efficiency of the new algorithm. The algorithm was first tested on generic one-dimensional (1D) state estimation problem, and then on bearings only tracking problem so as to provide a theoretical backing to novel state estimation method for object tracking. SMO-PF using DR [SMO (DR)-PF] was then tested on standard videos obtained from CAVIAR dataset, and indigenous video consisting of various object tracking challenges. The results were also obtained for simple PF, particle swarm optimisation assisted PF (PSO-PF), SMO-PF with normal resampling, and cuckoo search optimisation assisted PF (CS-PF) to provide comparative analysis.

The rest of the paper is organised as follows. Section 2 details some advancements made in the field of nature-inspired optimisation assisted PFs. Section 3 delineates the PF algorithm and SMO algorithm. In Section 4, the SMO (DR)-PF algorithm is discussed for 1D and 2D state estimation, and object tracking problem. Section 5 discusses the performance parameters used for evaluating the algorithms. In Section 6, results obtained from the new PF for the three problems are discussed in detail. Finally, the paper is concluded in Section 7.

2 Related work

Nature-inspired meta-heuristics approach to optimise sample distribution in the state hypothesis obtained from PF has been an area of active research in recent years. All nature-inspired optimisation methods share two common aspects: exploration and exploitation. Exploration is the ability of expanding search space, whereas exploitation is the ability of finding optima around a good solution [21]. First few iterations of optimisation algorithm are used for exploring the search space so as to avoid getting trapped in local optima. After some iterations, exploration dampens and exploitation surfaces and the system reaches a semi-optimal solution. In case of PF, the combination results in concentration of particles in high probability area aiding in creation of effective posterior hypothesis.

In [22], authors implemented ant colony optimisation (ACO) algorithm in PF to solve the sample impoverishment problem. ACO provides guaranteed convergence and can be used to optimise dynamic problems; time to convergence, however, is uncertain. A non-linear economic model was employed to test the performance of the algorithm. The results were also obtained for other improved PFs for comparison. The research, however, did not test the algorithm for bearings only tracking problem and object tracking. In [23], authors used artificial fish swarm optimisation (AFSO) for optimising particle distribution before resampling step so as to centralise particles in high-likelihood area. The algorithm was tested for manoeuvring target tracking. AFSO technique provides

high convergence speed, high accuracy, and fault tolerance. However, it has high time complexity associated with it [24]. In [25], authors proposed a genetic algorithm (GA) filter wherein they replaced the prediction step with mutation and crossover operators in GA. Though GA is fast in exploring the search space and has low memory requirements, the solution obtained by GA can be sub-optimal or, local maxima/minima.

PSO provides various advantages over many swarm-based optimisation techniques. The fitness function can be non-differentiable and PSO rapidly produces quality solution for problems involving many dimensions. However, since there is no general optimisation technique for practical and multidimensional problems, tuning and experimenting with input parameters of PSO method is necessary. Additionally, for noisy non-linear problems, stochastic variability of results obtained from the PSO method is very high. In [26], authors exploited capabilities of PSO and proposed a PF based on PSO to overcome particle deprivation problem, but its implementation did not cover object tracking problem. In [27], authors proposed a PSO assisted PF for object tracking in video sequences. They also exploited the ability of PF to fuse different cues and used the mixture of histogram of oriented gradients (HOG) and colour histogram to construct feature vector for target association. Later in [3], authors proposed an adaptive PSO assisted PF called APSO-PF to track manoeuvring object. The standard parameters in PSO-PF were made adaptive and were changed according to motion state of the object.

CS optimisation can effectively deal with multimodal problems and requires tuning of only single parameter. Walia and Kapoor [28] proposed a novel PF that embeds cuckoo search via levy flight to estimate the state of generic 1D state estimation problem and classic bearings only 2D state estimation problem. The new algorithm surpassed standard PF and PSO-PF in terms of root mean squared error (RMSE) and number of effective particles (NOEP). Later in [29], authors detailed the implementation of improved CS-based PF for object tracking. Comprehensive qualitative and quantitative analysis was done and the new algorithm was proved to be more reliable. In optimisation problems with high level of noise and non-linear fitness function, firefly optimisation algorithm (FA) outperforms PSO in finding the optimal solution and time taken to reach that optimum solution. In [30], authors proposed a novel PF based on FA to increase the number of meaningful particles for better approximation of state vector. However, FA sometimes get trapped in the local optima and is unable to completely get rid of it. Results were obtained for different video sequences consisting of various object tracking challenges. The new algorithm outperformed standard PF for tracking motile object. In [31], a general optimisation-based tracking architecture based on FA is proposed. The speed and accuracy of new FA-based tracking algorithm is compared with standard PF, meanshift, and PSO.

In [20], it was observed that SMO successfully overcame the problem of premature convergence and stagnation in the search space experienced by many state-of-the-art nature-inspired meta-heuristic optimisation techniques. SMO performed better than artificial bee colony optimisation, PSO, differential evolution, and covariance matrix adaption-evolution strategy for different benchmark problems.

3 Particle filtering and SMO

This section discusses the PF and SMO algorithm in detail. Additionally, the methodology of incorporating optimisation algorithm within a PF has been also expounded.

3.1 Particle filter

The key idea of particle filtering is to represent required PDF of state $p(x_t|Z_t)$, where $Z_t = \{z_1, z_2, \dots, z_t\}$ is the set of all available observations and x_t is the state estimate at time t , by set of random particles with associated weights. It is assumed that initial PDF $p(x_0|z_0)$ is known a priori.

Particle filtering consists of two steps:

- a. Prediction of state using motion model.
- b. Update state using measurement model.

These steps are applied recursively to obtain PDF $p(x_t|Z_t)$.

The prediction step uses the motion model $x_t = f_t(x_{t-1}, w_{t-1})$ to obtain the prior PDF at time t , provided $p(x_{t-1}|Z_{t-1})$ is known, where $Z_{t-1} = \{z_1, z_2, \dots, z_{t-1}\}$ is set of previously observed states up till time $t-1$. Here, f_t is a non-linear function of target state and w_{t-1} is zero-mean white noise sequence that is used to model unknown disturbances in the state prediction process. This model defines the translation, rotation, scale, and other state parameters of the target.

Online measurements are made according to model $z_t = h_t(x_t, u_t)$, here h_t is a non-linear function and u_t is a zero-mean white noise sequence. The measurement z_t forms the basis of the update step and is used to update the predicted state using the Bayesian rule as follows:

$$p(x_t|z_1:z_t) = \frac{p(z_t|x_t)p(x_t|z_1:z_{t-1})}{p(z_t|z_1:z_{t-1})} \quad (1)$$

$$p(z_t|z_1:z_{t-1}) = \int p(z_t|x_t)p(x_t|z_1:z_{t-1}) dx_t \quad (2)$$

As stated, particle filtering approximates the state by a weighted particle set $S = \{(x_t^{(n)}, w_t^{(n)}), n = 1, 2, \dots, N_s\}$. Here, N_s is equal to total number of particles. Each particle depicts one hypothetical state x_t of the target and has a weight w_t that is obtained in terms of degree of its association with the observation. These weights are normalised such that $\sum_{i=1}^{N_s} w_i^i = 1$. The phenomena of importance sampling comes into play to obtain this weighted approximation of posterior $p(x_t|Z_t)$. It states that, if $p_x \propto \pi(x)$ is a probability density through which it is difficult to obtain particles, but for which $\pi(x)$ can be evaluated, then a weighted approximation of the posterior PDF is given by

$$p(x) \simeq \sum_{i=1}^{N_s} w_i^i \delta(x - x^i) \quad (3)$$

Here, $w^i \propto (\pi(x^i)/v(x^i))$ is the normalised weight of the i th particle. $x \simeq v(x)$, $i = 1, 2, \dots, N_s$ are the samples that are generated from a proposal $v(\cdot)$ known as the importance density. These updated weighted approximation acts as prior for the next state and the process is repeated until last state is achieved. However, after several iterations most of the particle's weight is almost negligible and NOEP, as expressed in (4), rapidly decreases. Resampling was introduced to tackle this problem

$$N_{\text{eff}} = 1 / \sum_{i=1}^N (w_k^i)^2 \quad (4)$$

3.2 Spider monkey optimisation

SMO is a nature-inspired heuristic approach to solve optimisation problem. It is based on the foraging behaviour of SMs who fall under the category of fission-fusion social structure (FFSS) based animals [20]. The FFSS-based animals are social and live in groups of 40–50 individuals. The FFSS of swarm helps in reducing the competition amongst the group members by dividing them into smaller groups. A female global leader searches the food sources and divides the group into smaller sub-groups in case she is unable to procure sufficient food for the group. These smaller sub-groups then forage independently. These sub-groups also have a female local leader who decides an efficient foraging route every day. Members of these sub-groups communicate within and outside the sub-group about availability of food resources.

The SMO process used consists of three steps, and they are:

Step 1: Initialisation of population

In this step, N_s SMs, uniformly distributed, are generated. Each SM $\mathbf{SM}_n \{n = 1, 2, \dots, N_s\}$ is a D -dimensional vector, where D the number of state variables, and each $\mathbf{SM}_n \{n = 1, 2, \dots, N_s\}$ is a potential state.

Step 2: Local leader phase (LLP)

Here each SM \mathbf{SM}_n updates its position based on the information of the local leader and corresponding group member experience. After obtaining the new position, fitness value of that position is obtained. If new fitness value is higher than the previous fitness value, then \mathbf{SM}_n , member of the l th local group, updates its position with the new one according to the following equation:

$$\begin{aligned} \mathbf{SM}_{\text{new},nd} = & \mathbf{SM}_{nd} + U(0, 1) \times (\mathbf{LL}_{ld} - \mathbf{SM}_{nd}) + U(-1, 1) \\ & \times (\mathbf{SM}_{rd} - \mathbf{SM}_{nd}) \end{aligned} \quad (5)$$

\mathbf{SM}_{nd} is the d th dimension of n th SM, \mathbf{LL}_{ld} represents the d th dimension of the l th local group leader position, and \mathbf{SM}_{rd} is the d th dimension of r th SM that is chosen randomly from l th local group such that $r \neq n$, and $U(0, 1)$ is a uniformly distributed random number between 0 and 1. Fig. 1a shows the position update process used in LLP. L is the number of groups in total, N is the group size, and pr is the perturbation rate that controls the magnitude of perturbation in current state.

Step 3: Global leader phase (GLP)

All SMs update the position based on global leader experience and local group member experience. The position is updated according to the following equation:

$$\begin{aligned} \mathbf{SM}_{\text{new},nd} = & \mathbf{SM}_{nd} + U(0, 1) \times (\mathbf{GL}_d - \mathbf{SM}_{nd}) + U(-1, 1) \\ & \times (\mathbf{SM}_{rd} - \mathbf{SM}_{nd}) \end{aligned} \quad (6)$$

\mathbf{GL}_d represents the d th dimension of the global leader position and $d = \{1, 2, \dots, D\}$ is a randomly chosen index.

In this step, position of SMs (\mathbf{SM}_n) is updated according to probability p_n that is calculated according to the following equation:

$$p_n = \frac{0.9 \times \text{fitness}_n}{\text{max_fitness}} + 0.1 \quad (7)$$

Here, fitness_n is fitness of n th SM and max_fitness is the maximum fitness in the group. By assigning probability to a SM, it is made sure that good candidates have higher chances to make themselves better. Furthermore, new fitness values are compared with old fitness values and if new value is greater than old values, \mathbf{SM}_n updates its position. Fig. 1a also discusses the algorithm for GLP.

3.3 SMO-based PF

SMO (DR) algorithm is used in conjunction with PF to improve the efficiency of PF. First, the initial state PDF is obtained from the resampled particles of the previous time step. Next, motion model and measurement model (described in Section 4) are used to obtain the predicted PDF $p(x_t|z_t)$. With the help of a likelihood function, described in [4] ((25)), a weighted approximation of the state PDF, $p_1(x) \simeq \sum_{i=1}^{N_s} w_i^i \delta(x - x^i)$, is obtained wherein weight is assigned to a particle corresponding to the degree of its association (as measured with the measurement model) with the true state. After obtaining a weighted approximation of the state PDF, SMO algorithm is used to improve the weights of the particle. Every particle in the PF is treated as a SM. The set of particles with

Algorithm

```

/* Local Leader Phase */
for each  $l = \{1, 2, \dots, L\}$  do
    for each  $SM_n \{n = 1, 2, \dots, N\}$  do
        for each  $d = \{1, 2, \dots, D\}$  do
            if  $U(0, 1) \geq pr$  then
                 $SM_{new_{nd}} = SM_{nd} + U(0, 1) \times (LL_{ld} - SM_{nd}) + U(-1, 1) \times (SM_{rd} - SM_{nd})$ 
            else
                 $SM_{new_{nd}} = SM_{nd}$ 
            end if
        end for
    end for
end for
/* Global Leader Phase */
for each  $l = \{1, 2, \dots, L\}$  do
    count = 1;
    N= $l^{th}$  group size
    while count < N do
        for  $n = \{1, 2, \dots, N\}$  do
            if  $U(0, 1) < p_n$  then
                count = count + 1
                Randomly select  $d = \{1, 2, \dots, D\}$ 
                Randomly select  $SM_r$  from  $l^{th}$  group such that  $r \neq n$ 
                 $SM_{new_{nd}} = SM_{nd} + U(0, 1) \times (GL_d - SM_{nd}) + U(-1, 1) \times (SM_{rd} - SM_{nd})$ 
            end if
        end for
    if n is equal to N then
        n = 1;
    end if
end while
end for

```

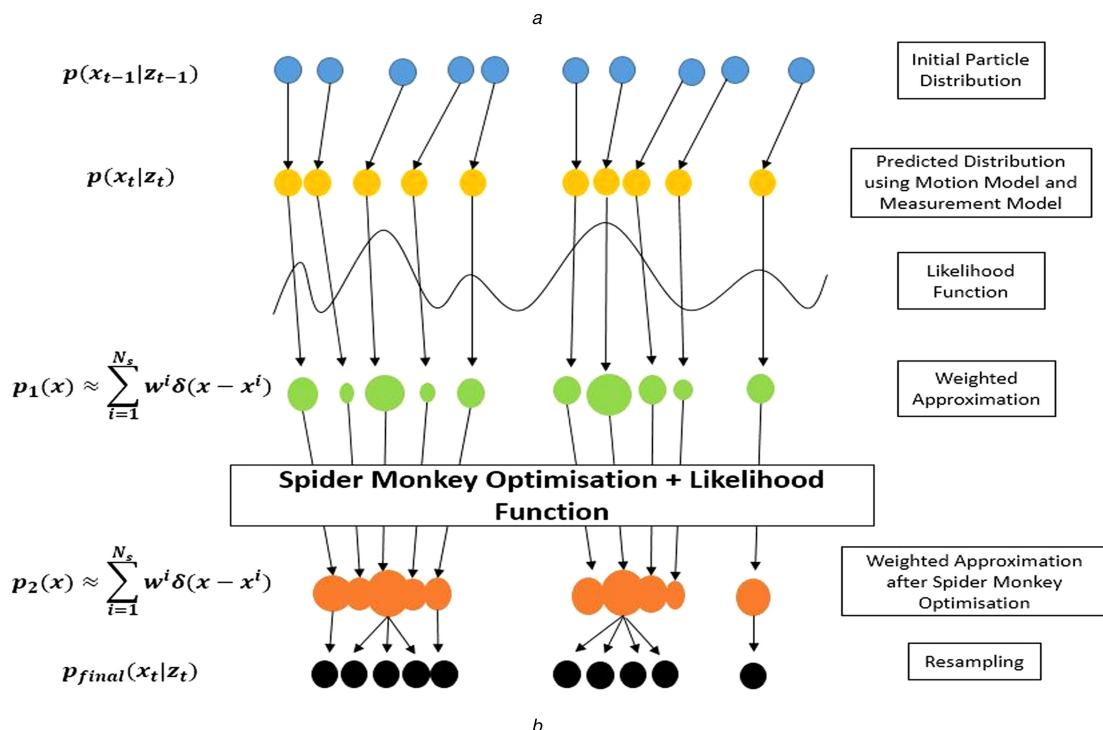


Fig. 1 SMO algorithm for object tracking

(a) Optimisation algorithm used to improve particle distribution, (b) Improved particle distribution using SMO

N_s SMs is divided into L local groups with each group having N SMs. The particle with highest fitness value/weight in the set is treated as the global leader and the particle with highest weight in each local group is treated as the local leader. Next, LLP phase is applied that helps the particles to explore the fitness space with high perturbation rate pr. This helps to overcome the problem of stagnation in the fitness space experienced by many optimisation techniques. After the LLP phase, GLP phase is applied that allows the particles with high weights to update their position and further improve their fitness values. As can be noticed from Fig. 1b, particles having low weights before SMO step have increased fitness values and are more close to the global leader. Also, the global leader has further improved its weight value. After obtaining weighted approximation via SMO, resampling step is applied to obtain the final PDF of the state. All in all, the optimisation step, in combination with DR, increases the diversity and quality of particles, and helps in effectively modelling the posterior PDF of the target.

4 One-dimensional state estimation, bearings only tracking, and object tracking by SMO (DR)-PF

In this section, SMO (DR)-PF algorithm used for generic 1D state estimation problem, bearing only tracking (2D state estimation) problem, and object tracking problem is discussed in detail. This section also discusses the motion models, measurement models and other implementation details.

4.1 One-dimensional state estimation problem and 2D state estimation problem

4.1.1 One-dimensional non-linear problem: The following non-linear model was considered to evaluate the performance of SMO (DR)-PF:

$$x_k = 0.5x_{k-1} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2(k-1)) + w_k \quad (8)$$

$$z_k = \frac{x_k^2}{20} + u_k \quad (9)$$

The above model was reported in [32]. It is a highly non-linear model, both in system (8) and the measurement (9) equation. Here w_k and u_k are zero-mean Gaussian white noise with variance 10.0 and 1.0, respectively.

4.1.2 Bearings only tracking problem: In this problem, the target's motion is governed by second-order model represented by

$$\begin{aligned} x_k &= \alpha x_{k-1} + \rho w_k \\ x_k &= (x, \dot{x}, y, \dot{y})_k^T \quad \text{and} \quad w_k = (w_x, w_y)_k^T \end{aligned} \quad (10)$$

Here, x and y denote Cartesian coordinates of the target and \dot{x} and \dot{y} represent velocity of the target. w_k denotes system noise and is zero-mean Gaussian white noise process with covariance $C: E[w_k w_j^T] = C\delta_{jk}$, where $C = cI_2$ and I_2 is 2×2 identity matrix. Measurement used to track target is bearing or angle z_k and is governed by measurement

$$z_k = \tan^{-1}\left(\frac{y_k}{x_k}\right) + u_k \quad (11)$$

Here, u_k is measurement noise and is zero-mean Gaussian white noise with variance $r: E[u_k u_j] = r\delta_{jk}$. Before taking measurements at $k=1$, the initial state vector is assumed to have a Gaussian distribution with known mean \bar{x}_1 and covariance $C_1 = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)$.

Fig. 2a presents the algorithm that has been implemented for 1D non-linear state estimation problem and bearings only tracking 2D state estimation problem.

4.2 SMO (DR)-PF for object tracking

This subsection discusses the algorithm implemented (Fig. 2b) for object tracking in standard videos, obtained from CAVIAR dataset [33], and synthetic videos.

First, the acquired frame is searched for the object to be tracked. This is done with the help of Gaussian mixture model (GMM) based foreground detection technique. In this method, intensity value of every pixel from the first few frames is modelled using mixture of Gaussian distributions. Aberrant pixel values that do not fit the background distribution are labelled as foreground. Number of Gaussian modes in mixture model used were 5. Algorithm presented in [34] was used to detect foreground object. Although GMM-based detection allows for some variability in the states of the pixel, the model assumes that this variability is derived from noise and not structured motion patterns such as swaying trees, moving clouds, and so on. Another drawback of the used technique is the lack of consistency between the states of adjacent pixels that often leads to abysmal foreground–background segmentation [35]. As will be noticed in Section 6.3, these shortcomings limit the efficiency achieved through the proposed object tracking framework. However, PF's ability to maintain multiple hypothesis of target's state helps in overcoming these drawbacks to some extent.

Thereafter, N particles are initialised to represent that state of object. After updating particles through (13), every particle is assigned a weight using the Bhattacharyya coefficient and weighted approximation of the state PDF is obtained. Next, these particles are divided into sub-groups for implementing the LLP and GLP of SMO algorithm. This step reduces degeneracy in the particle set and increases NOEP. Then the newly generated particles are resampled using DR. Finally, the state of the target is estimated by taking weighted mean of resampled particles. The loop is continued until all states are predicted.

4.2.1 Motion model: Motion of an object to be tracked can greatly vary from simple translation to complex rotation. Constant velocity model for translation motion and random walk model for the scaling and rotation have been considered. The used multicomponent model was proposed in [4].

The complete state vector is described as

$$\mathbf{x} = (x, \dot{x}, y, \dot{y}, \theta, s)^T \quad (12)$$

The state is updated according to the following equation:

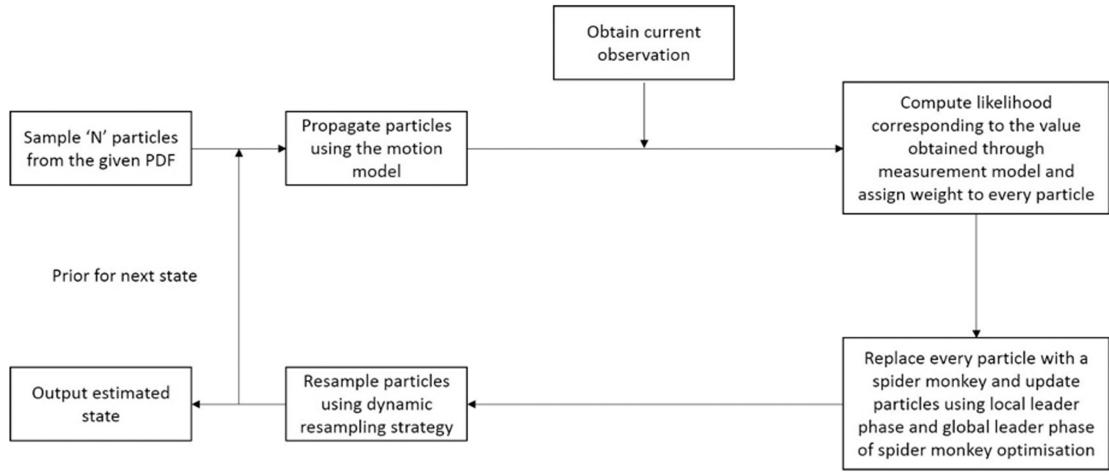
$$x_{k+1} = \mathbf{G}x_k + w_k, \quad w_k \sim N(0, \mathbf{C}) \quad (13)$$

$$\mathbf{G} = \begin{bmatrix} 1 & V & 0_{2 \times 2} & 0_{2 \times 1} & 0_{2 \times 1} \\ 0 & 1 & & & \\ 0_{2 \times 2} & \begin{bmatrix} 1 & V \\ 0 & 1 \end{bmatrix} & 0_{2 \times 1} & 0_{2 \times 1} & \\ 0_{1 \times 2} & 0_{1 \times 2} & 1 & 0 & \\ 0_{1 \times 2} & 0_{1 \times 2} & 0 & 1 & \end{bmatrix} \quad (14)$$

Here w_k is zero-mean Gaussian white noise, V is the sampling interval, and \mathbf{C} is the covariance matrix with covariance values for translation motion, rotation, and scaling as $\sigma_x^2, \sigma_y^2, \sigma_\theta^2$ and σ_s^2 , respectively.

4.2.2 Measurement model: After updating the state, every particle is assigned a weight. The weights are proportional to the degree of association of reference feature with the target feature.

This work uses colour histogram for computing weights of the particles, since colour histograms in general can accommodate small changes in the appearance of the object.



a

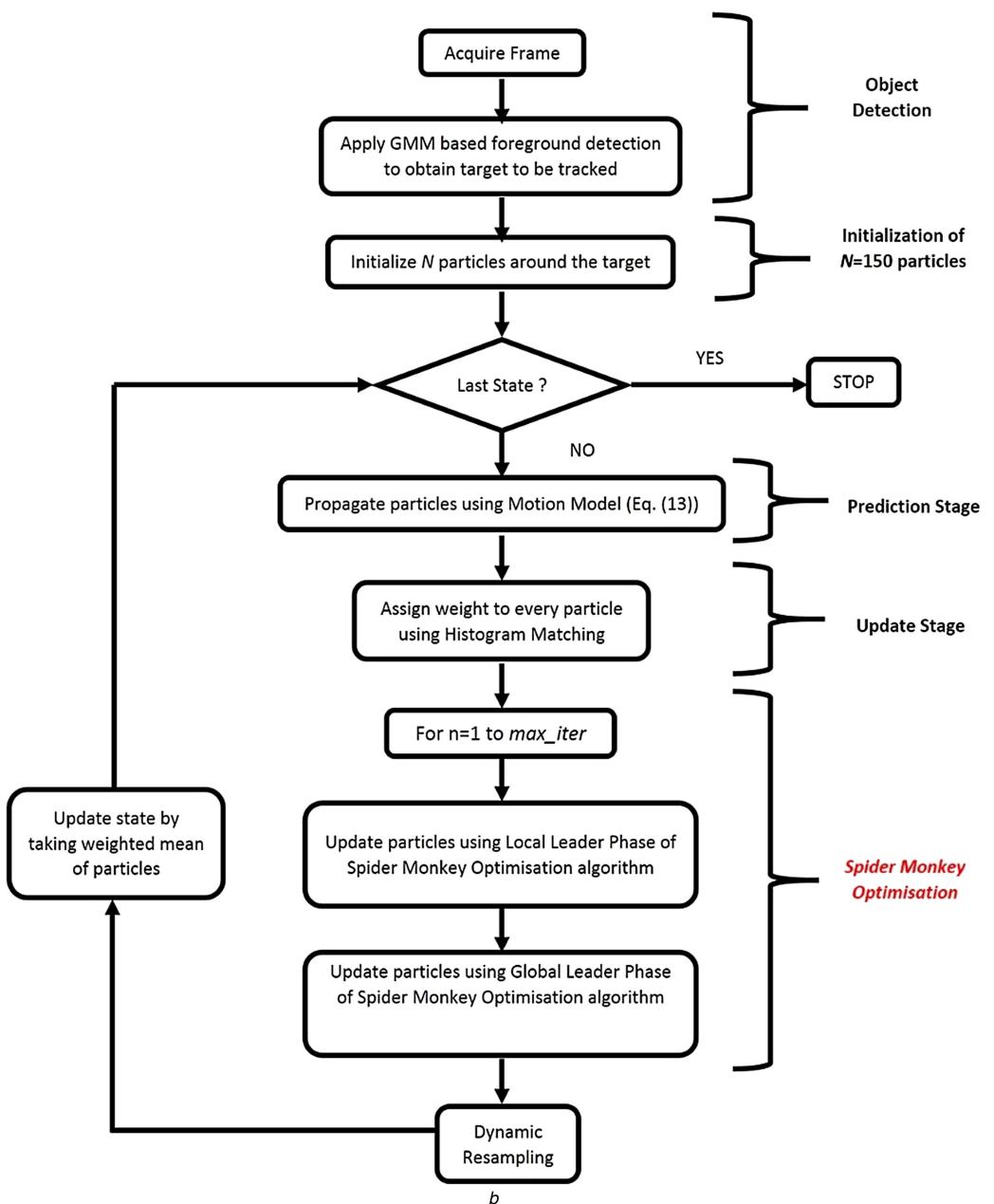


Fig. 2 SMO assisted PF algorithm

(a) For 1D and 2D state estimation problem, (b) For object tracking problem

Histogram, $h_t = (h_{1,t}, h_{2,t}, \dots, h_{n,t})$ for a region S_t corresponding to state t is given by

$$h_{i,t} = \sum_{u \in S_t} \delta_t(b_u), \quad i = 1, 2, \dots, n \quad (15)$$

Here, δ_i is the Kronecker-delta function at the bin index i . $b_u \in \{1, \dots, n\}$ is the histogram bin index associated the intensity at pixel location $u = (x, y)$ and n is the number of bins in every channel. Then histogram is normalised such that $\sum_i h_{i,t} = 1$. $8 \times 8 \times 8$ Bin RGB histogram was constructed.

After obtaining the target and reference histogram, similarity between the two histograms is measured. This manuscript uses Bhattacharyya distance for measuring similarity. Bhattacharyya distance (B.D.) has been previously used in [29, 30]

$$\text{B.D.} = \sqrt{1 - \text{B.C.}} \quad (16)$$

Here, B.C. is Bhattacharyya coefficient and is given by

$$\text{B.C.}(h_{\text{ref}}, h_{\text{tgt}}) = \sum_{i=1}^n \sqrt{h_{\text{ref},i} * h_{\text{tgt},i}} \quad (17)$$

Here, h_{ref} and h_{tgt} are the colour histogram of reference and target region, respectively. If the colour histograms are similar then B.C. is 1 indicating a perfect match.

Based on (16), distance D is defined which takes into consideration all three colour channels

$$D(h_{\text{ref}}, h_{\text{tgt}}) = \sqrt{\frac{\sum_{c \in \{R, G, B\}} \text{B.D.}^2(h_{\text{ref}}^c, h_{\text{tgt}}^c)}{3}} \quad (18)$$

Based on value of $D(h_{\text{ref}}, h_{\text{tgt}})$, weight is assigned to every particle.

5 Performance evaluation parameters

This section discusses the performance parameters used for evaluating the performance of the novel algorithm.

5.1 For 1D non-linear problem and bearings only tracking problem

A. *RMSE*: It is an excellent general purpose error measuring parameter and is defined as

$$\text{RMSE}_x = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad \text{For one - dimension} \quad (19)$$

Here, x_i is the true value of state and \hat{x}_i is the predicted value of the state. N is total number of estimated states

$$\text{RMSE}_{xy} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}{N}} \quad \text{For two - dimension} \quad (20)$$

Here, (x_i, y_i) are the true location coordinates of the target in $-xy$ plane, whereas (\hat{x}_i, \hat{y}_i) are the estimated location coordinates. N is total number of estimated states.

B. *Variance*: Results were obtained for 25 iterations where 1 iteration comprises 50 states for 1D non-linear problem, and 24 states for bearings only tracking problem. Variance depicts the stability of PF algorithm. Less the variance more stable is the performance of that algorithm.

C. *Time cost*: The total time taken by the algorithm to obtain the estimates of all states.

D. *NOEP*: ‘NOEP’ versus ‘state index’ graph was obtained to monitor the benefits brought by optimisation.

E. *Number of distinct particles (NODP)*: ‘NODP’ versus. ‘state index’ graph was obtained to monitor the benefits brought by DR.

5.2 For object tracking

A. *RMSE (in pixels)*: The combined RMSE error of pixel coordinate (x_i, y_i) to the estimated coordinate (\hat{x}_i, \hat{y}_i) was used and is defined as (20). $N = 10$ is equal to the total number of iterations for which the algorithm was run. This process is repeated for every frame. ‘RMSE’ versus ‘frame number’ graph is obtained.

B. *F-measure*: To calculate *F*-measure, it is paramount to understand recall and precision. Recall is the proportion of positive pixels that are correctly classified as target to be tracked

$$\text{Recall} = \frac{\text{number of correctly classified target pixels}}{\text{number of target pixels in GT}}$$

Precision is the proportion of predicted positive pixels that were correctly identified as target

$$\text{Precision} = \frac{\text{number of correctly classified target pixels}}{\text{number of target pixels estimated}}$$

Values of recall and precision are directly proportional to the performance of the algorithm; however, each performance evaluator can be misleading alone. Hence, their harmonic mean, *F*-measure was introduced to accommodate both.

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

C. *Success rate*: It is the percentage of total frames for which target object is accurately localised.

D. *Time cost*: It is the average amount of time (in seconds) taken to process one video frame of the video sequence to localise the target.

E. *NOEP*: Same as mentioned for 1D and 2D state estimation problem.

F. *NODP*: Same as mentioned for 1D and 2D state estimation problem.

6 Results and discussion

This section will cover the performance analysis of SMO (DR)-PF for 1D state estimation, 2D state estimation, and object tracking. Algorithms were implemented on MATLAB running on 1.4 GHz Dual Core Processor.

6.1 Generic 1D non-linear model

The initial state was assumed to be $x_0 = 0.1$. The PFs were initialised with prior PDF $p(x_0) = N(0, 2)$.

Fig. 3a shows the 50 states estimated through different PFs. Green line represents the true state obtained via (8), whereas black line represents the estimated state via PFs. Number of particles was equal to 500. Optimisation process brought an increase in efficiency by decreasing the difference between the estimated state and the true state. Figs. 3b and c display the NOEP and NODP at each state. Table 1 compares the mean RMSE for 25 iterations, variance of these RMSE values, and time cost for the five algorithms. Variance (RMSE) suggests increased stability of the proposed algorithms. It can be observed from Fig. 3c that the DR step brought a significant increase in the NODP than the systematic resampling technique used for other algorithms. Also, it can be observed from Fig. 3b that there is not much difference between NOEP for SMO-PF and SMO (DR)-PF. Hence, it can be concluded that SMO overcomes the particle degradation problem, while DR overcomes the particle impoverishment problem. Additionally, it can be observed from Table 1 that SMO (DR)-PF, though has higher RMSE than SMO-PF (perhaps because of the large measurement and process noise in the system), performs better than other improved PFs.

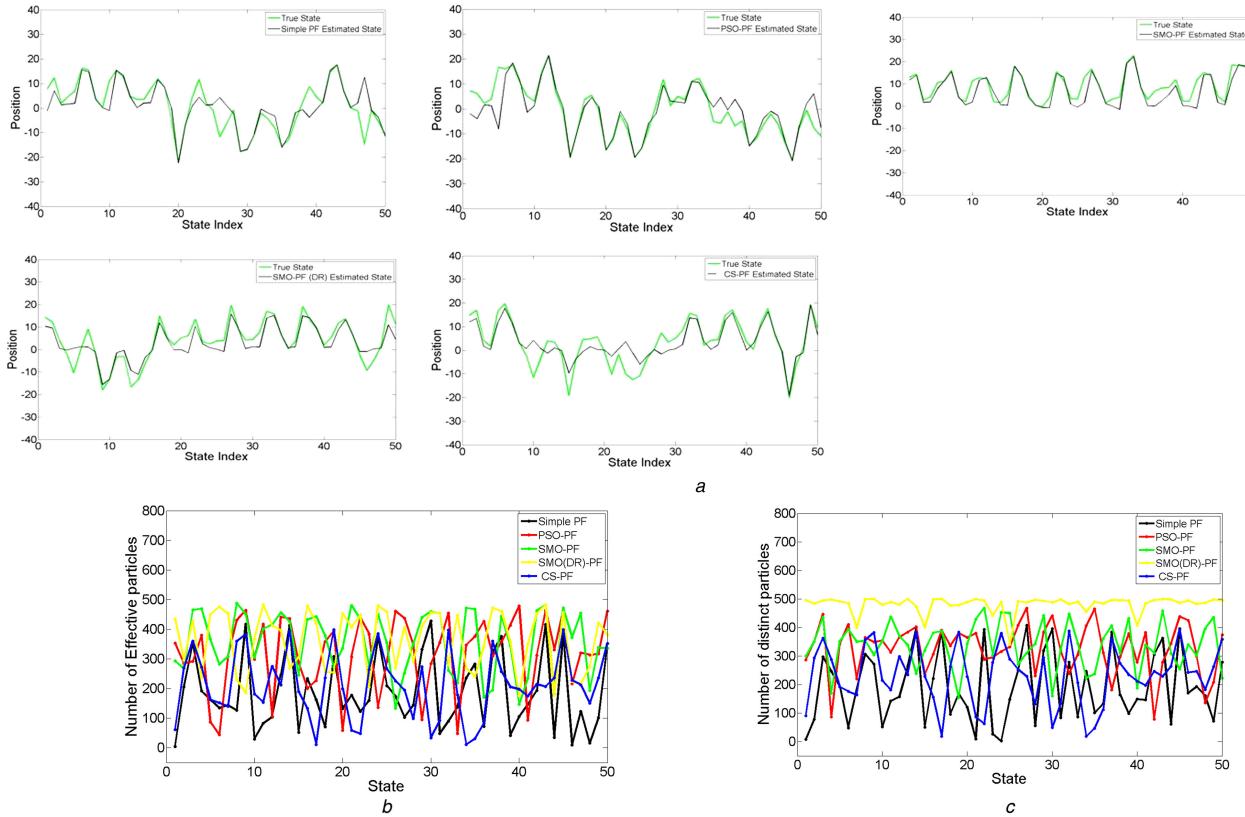


Fig. 3 Comparative performance evaluation

(a) State estimation using simple PF, PSO-PF, SMO-PF, SMO (DR)-PF, and CS-PF (left to right), (b) Number of effective particles, (c) Number of distinct particles

6.2 Bearings only tracking

The initial state of target was assumed to be $x_1 = (-0.05, 0.001, 0.7, -0.055)^T$. About 500 particles were taken to estimate the state. The prior distribution parameters were set to $\bar{x}_1 = (0.0, 0.0, 0.4, -0.08)^T$, and $\sigma_1 = 0.5$, $\sigma_2 = 0.005$, $\sigma_3 = 0.3$, and $\sigma_4 = 0.01$. The target trajectory is presented by red line and true trajectory by black line.

Fig. 4a presents the results at 24 states. It can be observed that SMO-PF closely follows the true trajectory, while others fail. Fig. 4b presents NOEP at 24 states. An interesting observation can be made from Fig. 4c. It can be seen that SMO (DR)-PF maintains excellent diversity in its particle set. On further exploring, it was found that SMO step alone increased the effective weight of particles above the threshold ($0.35 \times$ Total number of particles) and made the DR step redundant. Hence, no resampling step was required to increase effective weight of the particles. However, it can also be observed that RMSE of SMO (DR)-PF is more than SMO-PF. Table 2 compares the mean RMSE for 25 iterations, variance of these RMSE values, and time cost for the five algorithms. Low variance demonstrates stable performance of the proposed algorithm.

6.3 Object tracking problem

This sub-section discusses the performance of the proposed method. Algorithms were tested on three videos. Two videos were taken from the standard CAVIAR dataset and one video was

Table 1 Mean RMSE, variance, and time cost for all PFs (1D non-linear model)

	Mean RMSE	Variance (RMSE)	Time cost, s
Simple-PF	5.42	1.76	0.23
PSO-PF	4.92	1.39	0.84
SMO-PF	3.62	0.68	1.02
SMO(DR)-PF	4.65	0.52	0.96
CS-PF	4.84	1.12	0.78

synthesised indigenously. Number of particles N used to estimate the state of the target was set to 150. PFing infuses randomness in the object tracking process. To accommodate this randomness, every PF-based algorithm was run for ten times and results discussed are average of those observations. To maintain the even-handedness, same target model and motion model have been used for all algorithms, but normal resampling was used for other algorithms to exhibit benefits of DR. Table 3 reports the parameters used for tracking process.

6.3.1 OneLeaveShopReenter1cor.mpg: In this video, the person with dark red shirt is tracked. The person walks towards the camera thereby inducing scale transformation. During the course of the video, illumination in the scene varies substantially. The target object is occluded from frame 18 to frame 36. The video sequence incorporates shadowing challenge.

It can be noticed from Fig. 5d that SMO-PF maintained better F-measure than others. It can be further noticed that optimised PF (esp. SMO-PF) maintained higher NOEP (Fig. 5b) during occlusion. Fig. 5c present the total positional RMSE. As it can be noticed, positional RMSE for SMO-PF was lower than that for others (save few frames). Table 4 also compares the success rate and timing cost of the five algorithms. Fig. 5a presents some of the frames from the video sequence. Similar to the bearings only tracking problem, for SMO (DR)-PF, SMO alone overcame the particle degradation problem and retained the diversity (Fig. 5e).

6.3.2 ShopAssistant1cor.mpg: In this video, the person with white top is being tracked. The person walks away from the camera, thereby inducing scale changes. Change in velocity occurs in both directions. To make the sequence more challenging camouflaged background was used and the bottom of the target had significant similarity with clothes of the other person. The other person partially occluded the target in the frames 90–125. Additionally, as the target moves away, illumination changes and shadowing further complicate the task of tracking.

Fig. 6a presents some of the frames from the video sequence. Fig. 6b presents NOEP for all PFs. Fig. 6c presents the distance (in pixels) between the estimated centre and true centre. As it can be

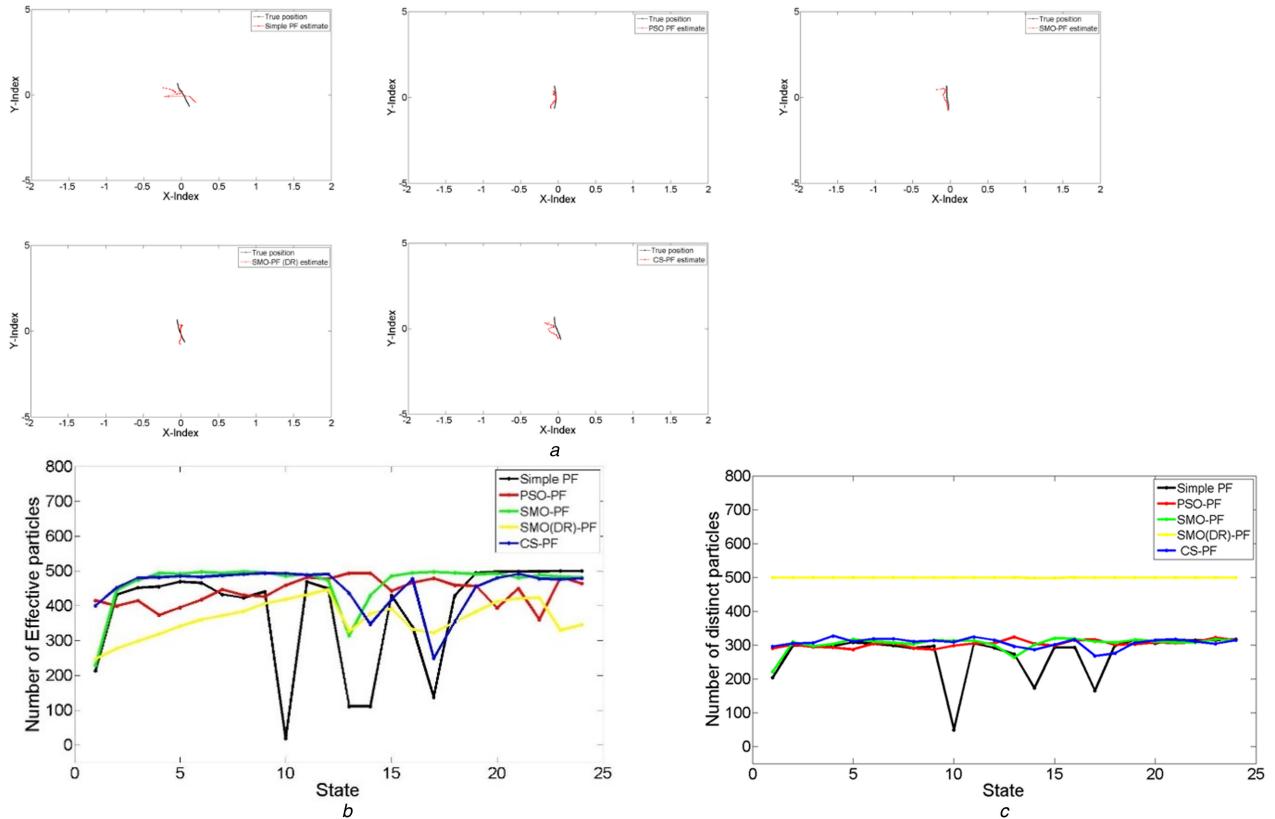


Fig. 4 Comparative performance evaluation

(a) State estimation using simple PF, PSO-PF, SMO-PF, SMO (DR)-PF, and CS-PF (left to right), (b) Number of effective particles, (c) Number of distinct particles

Table 2 Mean RMSE, variance, and time cost for all PFs (bearings only tracking)

	Mean RMSE	Variance (RMSE)	Time, s
Simple PF	0.20	0.0031	0.62
PSO-PF	0.16	0.00054	0.75
SMO-PF	0.06	0.00037	1.01
SMO(DR)-PF	0.14	0.00030	0.98
CS-PF	0.15	0.00045	0.69

Table 3 Features and parameters used for object tracking (bearings only tracking)

Number of particles	150
Number of bins for each histogram channel	8
$V, \sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_\theta^2$ for (13)	0.9, 0.04, 0.04, 0.0045, 0.0004
PSO parameters	$c1 = c2 = 2$, number of iterations = 3
SMO parameters	$pr = 0.8, L = 10, N = 15$, number of iterations = 2
DR parameters	$N_{th} = 0.35 \times \text{number of particles}$
CS parameters	P (discovery of alien eggs) = 0.8, number of iterations = 2
Processor	1.4 GHz dual core
Camera speed	23 fps

noticed, this distance is very less for SMO-PF and SMO (DR)-PF when compared with others. Fig. 6d shows the F -measure for all algorithms. As evident from these figures, overall performance of SMO-PF transcends over other PFs; however, performance of SMO (DR)-PF is very close to that of SMO-PF and in few frames even better. Additionally, in SMO (DR)-PF, SMO alone overcomes the particle degradation problem and retains the diversity in the particle set (Fig. 6e). Hence, resampling step became redundant. This helped to reduce the timing cost associated with SMO-PF as can be noticed from Table 5.

6.3.3 2StudentsConsiderableRotation.mpg: In this video, the person with the black shirt is being tracked. As the person moves from left to right, there are slight scale changes. The resolution was kept low and the colour of target's clothes was not significantly different from the background. Waving trees in the background added to the complexity in tracking the target. From frames 48 to 56, the target was occluded by another person. From Frame 93 to 105, the target rotated significantly. Fig. 7a presents some of the frames. It can be seen that simple PF failed to capture the rotation of target in these frames. On the other hand, SMO (DR)-PF, SMO-PF, CS-PF, and PSO-PF initially struggled to capture the rotation but eventually they did (Figs. 7c and d). It is interesting to note that NOEP maintained by PSO-PF is, on average, better than CS-PF for every video. Higher number of iterations used for PSO-PF can be the cause of these results. Additionally, SMO alone handled the particle degeneracy and impoverishment issue (Figs. 7b and e).

Table 6 discusses the timing cost and success rate of the five algorithms. Though timing cost for SMO-PF and SMO (DR)-PF is higher than the others, the higher success rate of the proposed algorithms ensures improved performance in applications where efficiency is primary and timing is secondary.

7 Conclusion

This paper presents a PF based on SMO and DR for object tracking. The SMO (DR)-PF implements SMO to overcome particle degradation problem and DR strategy to overcome the particle impoverishment problem caused by the sub-optimal resampling strategies. SMO strategy is a nature-inspired optimisation algorithm that emulates the foraging behaviour of a group of SMs. A group of SMs consists of one global female leader (having maximum fitness), and is divided into smaller groups. Each group has a local leader having maximum fitness value in the local group. Every SM updates its position with the help of the experience of local and global leader, thereby improving its fitness value. Using this line of thought in PF process results in improved quality of the particles in the particle set. DR strategy was employed to overcome particle impoverishment problem in PFs. For object tracking and bearings only tracking, it is also observed that SMO alone

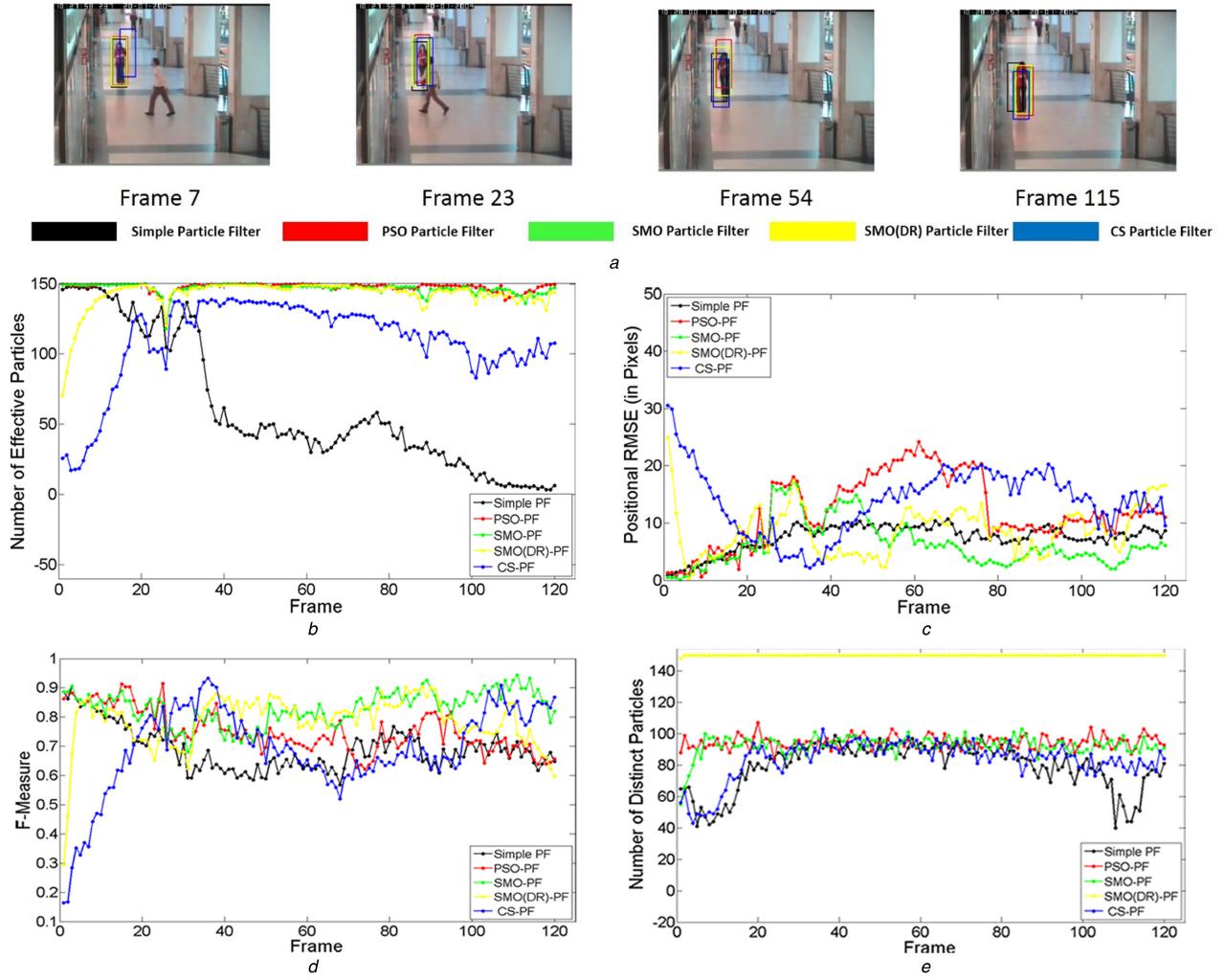


Fig. 5 Comparison of performance for five PFs on the basis of

(a) Sample frames from OneLeaveShopReenter1cor.mpg, (b) Number of effective particles, (c) Positional RMSE (in pixels), (d) F-measure, (e) Number of distinct particles

Table 4 Time cost and success rate for OneLeaveShopReenter1cor.mpg

	Time cost, s	Success rate, %
Simple PF	0.15	64.17
PSO-PF	0.31	92.50
SMO-PF	0.36	99.17
SMO(DR)-PF	0.35	98.33
CS-PF	0.30	90.00

increases the effective weight of particles and obviates the need of resampling step. The proposed PF is tested on standard videos obtained from the CAVIAR dataset and indigenously synthesised video containing various object tracking challenges. Proposed PF is also used to estimate state of non-linear 1D model and bearings only tracking problem. SMO (DR)-PF is compared with simple PF, SMO-PF based on normal resampling, CS-PF, and PSO-PF for the above mentioned state estimation problems. Though the SMO-PF and SMO (DR)-PF have slightly higher timing cost than simple-PF, PSO-PF, and CS-PF, they use fewer number of iterations than PSO-PF to converge to an optimum solution. It is interesting to note that SMO-based PFs perform better than CS-based PF for same number of iterations. Since, no direct comparison between SMO and CS exist in literature, it is a subject of discussion whether SMO is better than CS optimisation on benchmark functions. On the basis of results obtained through various performance evaluation parameters, it can be concluded that the SMO-based PFs perform better than other improved PFs for object tracking. This work will be extended to track multiple targets and more cues will be added to track those targets.

8 Acknowledgments

The authors thank anonymous reviewers for their valuable comments to improve the quality of this research.

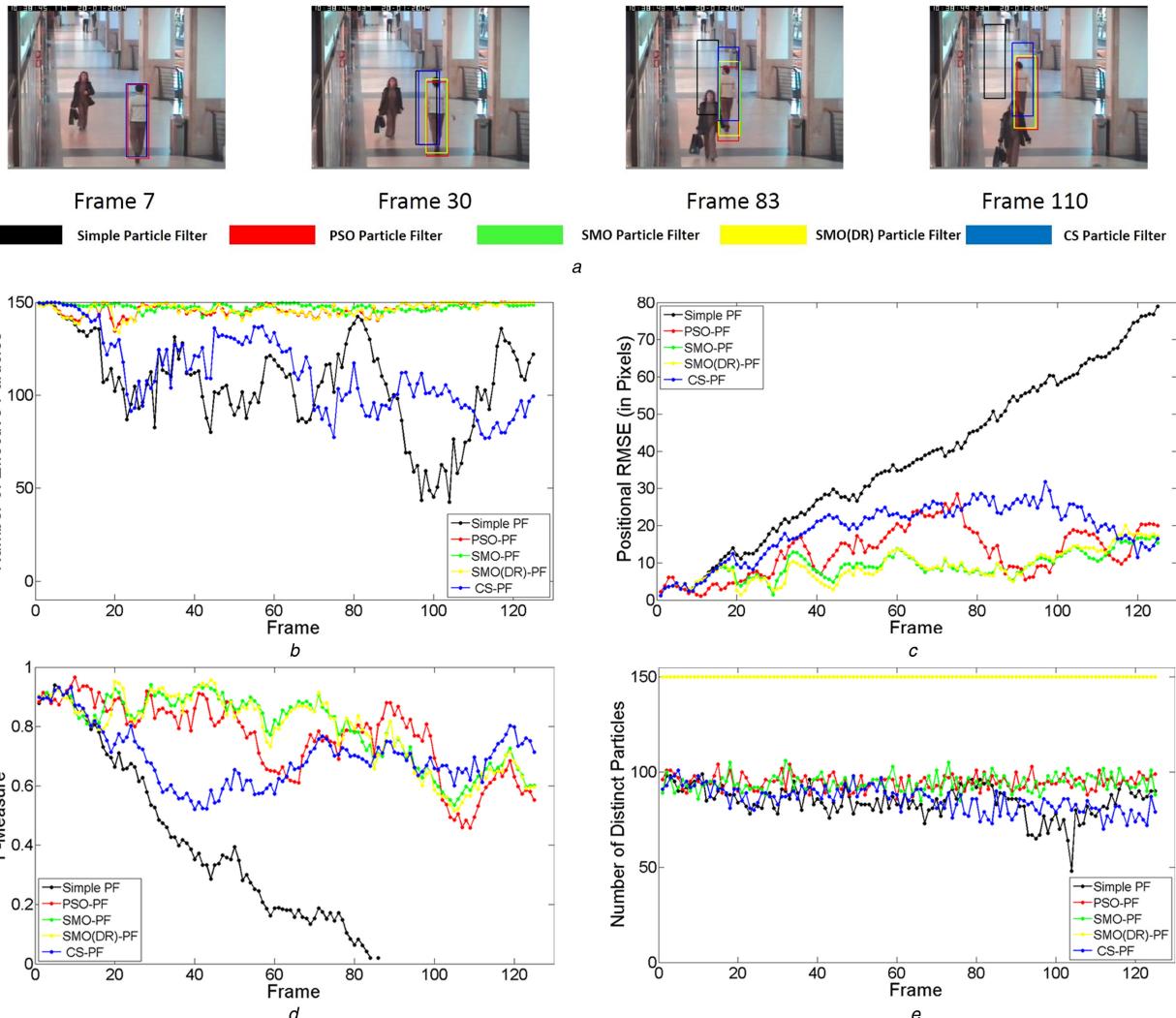


Fig. 6 Comparison of performance for five PFs on the basis of

(a) Sample frames from ShopAssistant1cor.mpg, (b) Number of effective particles, (c) Positional RMSE (in pixels), (d) F-measure, (e) Number of distinct particles

Table 5 Time cost and success rate for ShopAssistant1cor.mpg

	Time cost, s	Success rate, %
Simple PF	0.18	20.00
PSO-PF	0.46	79.20
SMO-PF	0.48	84.00
SMO(DR)-PF	0.47	80.00
CS-PF	0.45	68.44

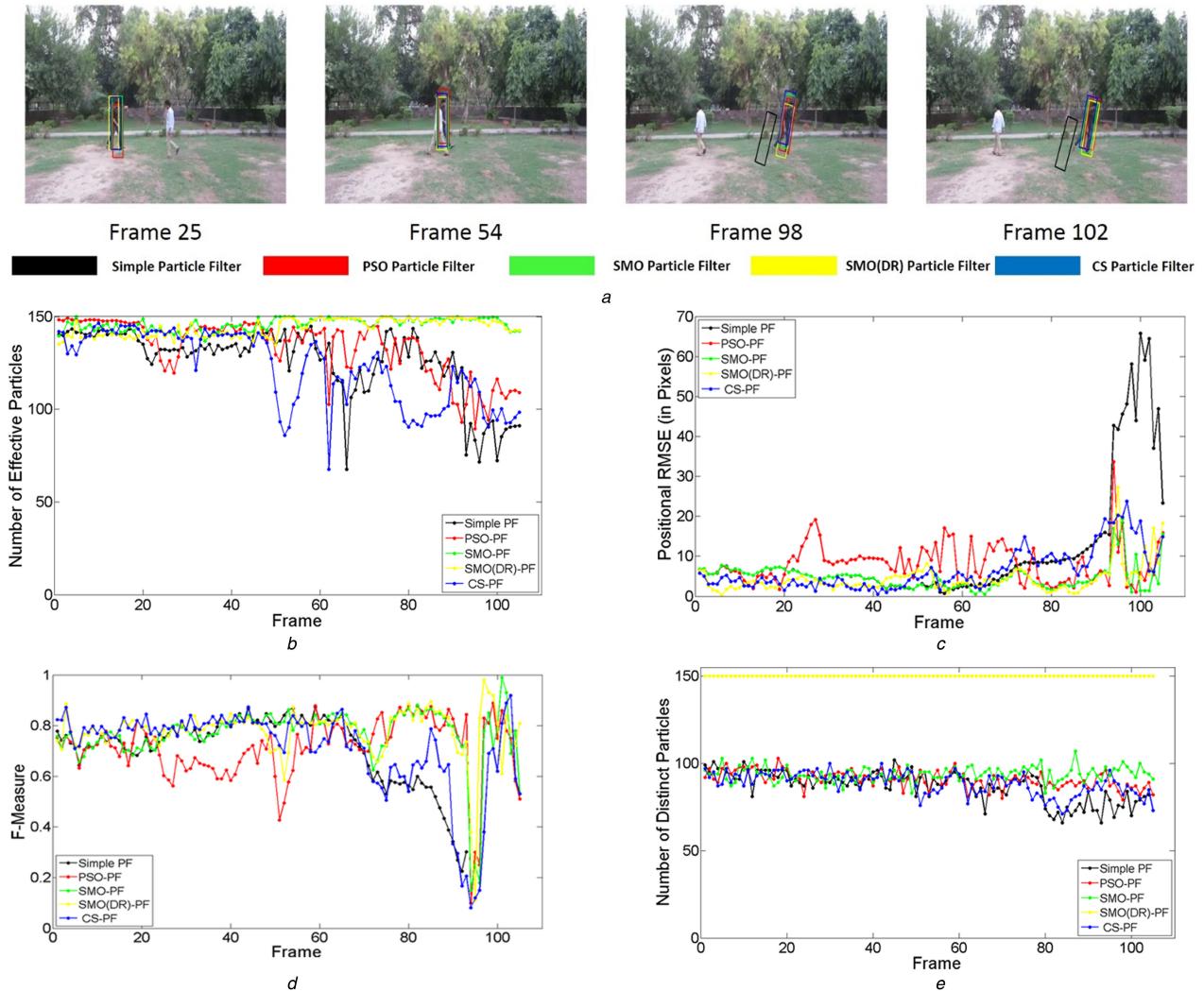


Fig. 7 Comparison of performance for five PFs on the basis of

(a) Sample frames from 2StudentsConsiderableRotation.mpg, (b)Number of effective particles, (c) Positional RMSE (in pixels), (d) F-measure, (e) Number of distinct particles

Table 6 Time cost and success rate for 2StudentsConsiderableRotation.mpg

	Time cost, s	Success rate, %
Simple PF	0.26	65.71
PSO-PF	0.49	80.95
SMO-PF	0.58	95.24
SMO (DR)-PF	0.55	96.19
CS-PF	0.44	75.24

9 References

- [1] Pan, X., Chen, X., Men, A.: ‘Occlusion handling based on particle filter in surveillance system’. 2nd Int. Conf. on Computer Modelling and Simulation, Hainan, 2010, pp. 179–183
- [2] Bradski, G.R.: ‘Real time face and object tracking as a component of a perceptual user interface’. IEEE Workshop on Applications of Computer Vision, Princeton, 1998, pp. 214–219
- [3] Qie, Z., Li, J.X.: ‘Adaptive particle swarm optimization-based particle filter for tracking maneuvering object’. Chinese Control Conf., Nanjing, 2014, pp. 4685–4690
- [4] Brasnett, P., Mihaylova, L., Bull, D., et al.: ‘Sequential Monte Carlo tracking by fusing multiple cues in video sequences’, *Image Vis. Comput.*, 2007, **25**, (8), pp. 1217–1227
- [5] Babenko, B., Yang, M., Belongie, S.: ‘Visual tracking with online multiple instance learning’. IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2009, pp. 983–990
- [6] Kalal, Z., Mikolajczyk, K., Matas, J.: ‘Tracking-learning-detection’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (7), pp. 1409–1422
- [7] Gordon, N., Salmond, D., Smith, A.: ‘A novel approach for nonlinear/non-Bayesian state estimation’, *IEE Proc. F, Radar Signal Process.*, 1993, **140**, (2), pp. 107–113
- [8] Isard, M., Blake, A.: ‘Condensation – conditional density propagation for visual tracking’, *Int. J. Comput. Vis.*, 1998, **29**, (1), pp. 5–28
- [9] Crisan, D., Moral, P., Lyons, T.: ‘Discrete filtering using branching and interacting particle systems’, *Markov Process. Relat. Fields*, 1999, **5**, (3), pp. 293–319
- [10] Arulampalam, M., Maskell, S., Gordon, N., et al.: ‘A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking’, *IEEE Trans. Signal Process.*, 2002, **50**, (2), pp. 174–188
- [11] Thrun, S., Burgard, W., Fox, D.: ‘Probabilistic robotics’ (The MIT Press, 2005)
- [12] Carpenter, J., Clifford, P., Fearnhead, P.: ‘An improved particle filter for nonlinear problems’, *IEEE Proc. Radar Sonar Navig.*, 1999, **146**, (1), pp. 2–7
- [13] Wang, Z., Liu, Z., Liu, W., et al.: ‘Particle Filter algorithm based on adaptive resampling strategy’. IEEE Conf. on Electronic and Mechanical Engineering and Information Technology, China, 2011, pp. 3138–3141

- [14] Liu, J.S., Chen, R.: ‘Sequential Monte Carlo methods for dynamical systems’, *J. Am. Stat. Assoc.*, 1998, **93**, pp. 1032–1044
- [15] Efron, B., Tibshirani, R.: ‘*An introduction to the bootstrap*’ (Chapman & Hall, 1993)
- [16] Zuo, J.: ‘Dynamic resampling for alleviating sample impoverishment of particle filter’, *IET Radar Sonar Navig.*, 2013, **7**, (9), pp. 968–977
- [17] Choe, J., Wang, T., Liu, F., et al.: ‘Particle filter with spline resampling and global transition model’, *IET Comput. Vis.*, 2014, **9**, (2), pp. 184–197
- [18] Zhai, J., Yeary, M.: ‘Implementing particle filters with Metropolis–Hastings algorithms’. Region 5 Conf.: Annual Technical and Leadership Workshop, 2004, pp. 149–152
- [19] Van Der, R., Doucet, A., De Freitas, N., et al.: ‘The unscented particle filter’. NIPS, 2000, pp. 584–590
- [20] Bansal, J., Sharma, H., Jadon, S., et al.: ‘Spider monkey optimization algorithm for numerical optimization’, *Memet. Comput.*, 2014, **6**, (1), pp. 31–47
- [21] Rashedi, E., Nezamabadi, H., Saryazdi, S.: ‘GSA: a gravitational search algorithm’, *Inf. Sci.*, 2009, **179**, (13), pp. 2232–2248
- [22] Zhong, J., Fung, Y.: ‘A biological inspired improvement strategy for particle filter’. IEEE Conf. on Industrial Technology, Gippsland, February 2009, pp. 1–6
- [23] Liang, X., Feng, J., Li, Q., et al.: ‘A swarm intelligence optimization for particle filter’. Proc. of World Congress Intelligent Control and Automation, Chongqing, 2008, pp. 1986–1991
- [24] Neshat, M., Sepidnam, G., Sargolzaei, M.: ‘Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications’, *Artif. Intell. Rev.*, 2014, **42**, (4), pp. 965–997
- [25] Higuchi, T.: ‘Monte Carlo filter using the genetic algorithm operators’, *J. Stat. Comput. Simul.*, 1997, **59**, (1), pp. 1–23
- [26] Tong, G., Fang, Z., Xu, X.: ‘A particle swarm optimized particle filter for nonlinear system state estimation’. IEEE Congress Evolutionary Computation, Vancouver, 2006, pp. 438–442
- [27] Zhao, J., Li, Z.: ‘Particle filter based on particle swarm optimization resampling for vision tracking’, *Expert Syst. Appl.*, 2010, **37**, (12), pp. 8910–8914
- [28] Walia, G.S., Kapoor, R.: ‘Particle filter based on Cuckoo search for nonlinear state estimation’. Int. Advance Computation Conf., India, 2013, pp. 918–924
- [29] Walia, G.S., Kapoor, R.: ‘Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search’, *Expert Syst. Appl.*, 2014, **41**, (14), pp. 6315–6326
- [30] Gao, M., Li, L.-L., Sun, X., et al.: ‘Firefly algorithm (FA) based particle filter method for visual tracking’, *Optik*, 2015, **126**, (18), pp. 1705–1711
- [31] Gao, M.L., He, X.H., Luo, D.S., et al.: ‘Object tracking using firefly algorithm’, *IET Comput. Vis.*, 2013, **7**, (4), pp. 227–237
- [32] Kitagawa, G.: ‘Non-Gaussian state-space modelling of nonstationary time series’, *J. Am. Stat. Assoc.*, 1987, **82**, (400), pp. 1032–1063
- [33] ‘CAVIAR Test Case Scenarios’. Available at <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA/>, accessed 15 May 2016
- [34] Stauffer, C., Grimson, W.: ‘Adaptive background mixture models for real-time tracking’. IEEE Computer Society Conf. on Computer Vision Pattern Recognition, Fort Collins, June 1999, pp. 246–252
- [35] Chan, A.B., Mahadevan, V., Vasconcelos, N.: ‘Generalized Stauffer–Grimson background subtraction for dynamic scenes’, *Mach. Vis. Appl.*, **22**, (5), 2011, pp. 751–766