International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

Particle Filter Based Visual Tracking: A Review

Rachna Verma[1] and Arvind Kumar Verma[2]
[1] *Department of Computer Science and Engineering, JN Vyas University, Jodhpur, Rajasthan, India.*
[2] *Department of Production and Industrial Engineering, JN Vyas University, Jodhpur, Rajasthan, India.*
Corresponding author: Rachna Verma

Submitted: 15-07-2021 Revised: 29-07-2021 Accepted: 31-07-2021

ABSTRACT: Particle Filter is one of the widely used techniques in visual tracking as it can model a dynamic environment with non-linear motions and multimodal non-Gaussian noises. Many decades of active research in visual tracking using particle filter has improved its various techniques, such as importance proposal, particle degeneracy and impoverishment, parallel implementation of resampling and weight update, data association, and target labelling, to make particle filter more accurate and efficient. In the last decade, many attempts have been reported which integrate the particle filter with the convolutional neural network. This integration has produced more accurate visual trackers as compared to traditional particle filter-based techniques. However, there are many unresolved challenges, such as variations in illumination, rapid and sudden change in motion, deformation of targets, complex and cluttered dynamic environment that need further research. Multiple target tracking is posing additional problems, such as identification and labeling of targets, track model drifting, misdetection, and computational explosion with the increase in the number of targets. In this paper, a review of recent advances, specifically in the last decade, in single target tracking and multiple targets tracking using particle filter is reported and research gaps are identified to give impetus to further research.

KEYWORDS: Particle Filter, state estimation,single-target tracking, multiple targets tracking, visual tracking

1.INTRODUCTION

Visual tracking, a sub-discipline of computer vision, aims to track moving objects, called targets, in a series of video frames. Given a sequence of images, visual tracking aims to estimate the location, shape, motion trajectory, and size of objects over time[1]. It is a hot area of research in computer vision as it has many practical applications, such as smart video surveillance, automatic driving, robot navigation, human-computer interaction, event analysis, etc. Humans and animals track moving objects effortlessly, but this simple task is eluding computer vision researchers for many decades.

Visual tracking involves the following four processes: initializing targets, modeling the appearance of the targets, predicting the motion of the targets, and positioning the targets in the next frame. Usually, targets are initialized either by manually marking bounding boxes around the targets in the initial frame or received as the output of an object detector. Target appearance modeling involves extracting characteristic visual features of the targets that differentiate them from the background and among each other. Extracted features are vital for the performance of a tracker. The two approaches used to extract features are hand-crafted features and deep learning features. In hand-crafted features, commonly used features are colors, texture, histograms, histograms of oriented gradients, edges, motion, etc. Recently, deep learning has proved its potential to detect and model objects by automatically learning relevant features during its training. Many works have been reported in visual tracking that use deep learning to model the appearance of targets. In motion prediction, the target location is estimated in the next frame using its current velocity and other parameters. It is a systematic attempt to increase the likelihood of finding targets in some specific areas of the frame in place of searching for the targets in the whole frame. Finally, in the positioning process, the predicted positions of the targets are verified and adjusted based on the available measurements.

The previous few decades have seen many visual tracking algorithms developed by researchers but with limited performance due to the myriad of challenges, such as background clutter, illumination variations, full and partial occlusions, scale variation, low resolution, fast motion, sudden change in direction, out of view, motion blur, deformation, object rotation, etc. Visual tracking approaches developed can be broadly categorized as feature-based, segmentation-based, estimation-based, and learning-based methods [2]. In the

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

feature-based approach, unique and characteristic features of targets, such as color, texture, optical flow, etc., are used to model targets and used to find the most similar objects in the next frame.

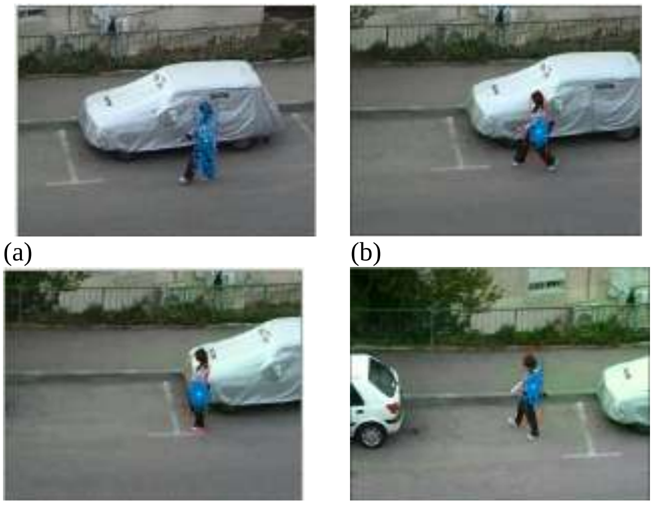
In the segmentation-based approach, foreground objects (targets) are separated from the background scene by segmenting each frame of the video. Once separated, the objects in the current frame are matched with the targets in the previous frame using some distinguishing features.

In the estimation-based approach, visual tracking is formulated as a state-space estimation problem, where a target is represented by a state vector consisting of its position and velocity. This formulation makes it very suitable to use the general Bayesian filters to continuously predict and update the position of the target based on the latest sensor data. This approach is one of the most widely researched methods, specifically particle filter-based tracking, of visual tracking and is the focus of the current paper.

In the learning-based approach, the appearance model of a target is created by automatically learning the features of the target using a machine learning technique. The learned appearance model is used to detect the target in subsequent frames. Machine learning techniques, especially deep learning, have proved to be very useful in accurately detecting and classifying objects in images. Many works have been reported using this technique (refer [3]), but this approach is computationally expensive and requires a lot of training datasets. To make better utilization of the discriminative power of the learned appearance model, many researchers have integrated a learning-based approach with a particle filter-based approach. This combination has given many state-of-the-art visual trackers with real-time performance. This paper also reviewed work done with the combination of deep learning and particle filter(PF).

This review paper is primarily focused on recent advances in PF-based visual tracking in the last decade. PF [4]-[6] is a state estimation method for a system with a non-linear process and measurement model corrupted with noise which may be non-Gaussian and multimodal. It is a tool to perform statistical inference of posterior distribution via Bayesian filtering in a state-space model. In a Bayesian approach to dynamic state estimation, the posterior probability density function (PDF) is constructed by a set of random samples, also called particles, with associated weights, and computes the state estimation from these particles and their weights. Further, the PDF is maintained by iteratively updating the state and weights of these particles.

PF has been successfully applied in various computer vision tasks, such as medical image analysis[7], image segmentation[8], image reconstruction [9]. It was first introduced in visual tracking by Isard and Blake [10] as the Condensation algorithm which tracks a single object in clutter.Visual tracking using PF has been used in vehicle tracking [11], face tracking [12], [13], hand tracking [14], robot localization [15], etc. Figure 1 shows a few frames (OTB2013 dataset [16]) in which a target is tracked using PF. In the figure, all particles converge as the filter models the target motion, and the mean state of the particles represents the target.



(a) (b) (c) (d)

Figure 1: Visual tracking using PF on Woman sequence of OTB2013 dataset, (a) particle initialization, (b-d) woman tracked using particle filter on different frames of a video (particles shown in blue color).

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

1.1 MOTIVATION AND CONTRIBUTION

The last few decades have seen a lot of research in visual tracking using PF and reported many research papers. Rao and Satyanarayana [17] surveyed various PF-based tracking methods developed till 2013. The review by Wang et al. [18] focuses on remarkable achievements in single-target tracking and discusses various challenges in multi-target tracking. This review paper presents the state-of-the-art of PF-based visual tracking research. We primarily reviewed the papers published during the last ten years for both single target tracking (STT) and multiple target tracking (MTT) using PF. Recently, some works have been reported that combine PF and deep learning for visual tracking. The results obtained by this combination are superior to traditional PF methods. We have attempted to summarise the works that cover the combination of deep learning and PF which is lacking in previous survey papers. The contributions of this paper are:

- a comprehensive review of PF-based tracking for STT and MTT for the last decade is provided.
- a review of integrated PF and deep learning-based visual tracking systems is presented.
- Research challenges, open issues, and research gaps are identified for further research.

Section 2 presents the overview of PF. Research issues and current trends in PF for STT and MTT are discussed in section 3 which also covers deep learning and PF for visual tracking. Section 4 describes various tracking benchmark datasets referred to in this paper, and Section 5 covers future research directions followed by the conclusion.

II.PARTICLE FILTER

2.1 OVERVIEW OF PARTICLE FILTER

In a Bayesian framework for tracking, PF estimates state sequence x_1, x_2, ... x_t from a sequence of measurements (data) z_1, z_2, ...z_t, using two basic equations: the state equation and the measurement equation. The state equation (Eq. (1)) describes the evolution of the state with time, whereas, the measurement equation (Eq. (2)) describes the relationship between the noisy measurements and the state.

$$x_t = f_t(x_{t-1}, u_{t-1}) \quad (1)$$

$$z_t = g_t(x_t, v_t) \quad (2)$$

Where $f_t: R^{d_x} \times R^{d_u} \to R^{d_x}$ and $g_t: R^{d_x} \times R^{d_v} \to R^{d_z}$ are nonlinear functions, and u_{t-1} and v_t represent the process noise and measurement noise, respectively.

From a Bayesian perspective, the tracking problem is to construct the posterior probability density function, $p(x_t|z_{1:t})$, of the state using all of the measurements up to time t, and the previous PDF $p(x_{t-1}|z_{1:t-1})$ available at time t. It is assumed that the initial PDF $p(x_0|z_0) \equiv p(x_0)$ of the state vector is available; $p(x_0|z_0)$ is known as the prior [6].

To obtain the posterior PDF, two stages are used: prediction and update [6]. The prediction stage uses the system model (Eq. (1)) to get the prior PDF of the state at time t using equation (Eq. (3))

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})\, p(x_{t-1}|z_{1:t-1}) dx_{t-1} \quad (3)$$

At time t when measurement z_t is available, the update stage modifies the prior PDF to obtain the required posterior PDF, $p(x_t|z_{1:t})$, of the current state using equation (Eq.(4))

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)\, p(x_t|z_{1:t-1})}{\int p(z_t|x_t)\, p(x_t|z_{1:t-1})\, dx_t} \quad (4)$$

Where $p(z_t|x_t)$ is the likelihood function, defined by the measurement model (Eq. (2)) and the known statistics of v_t.

Unfortunately, the general analytical solutions of Eq. (3) and Eq. (4) are not available for nonlinear non-Gaussian systems. However, PF provides a discrete weighted approximation of $p(x_t|z_{1:t})$. It approximates the state posterior $p(x_t|z_{1:t})$ as a set of weighted particles $\{x_t^i, w_t^i\}_{i=1}^{N_s}$ that evolves through time, as given in Eq. (5):

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i\, \delta(x_t - x_t^i) \quad (5)$$

Where x_t^i, $i = 1,..,N_s$ are the particles with associated weights w_t^i at time t, and $\delta(.)$ is the Dirac delta function. The weights are positive and sum to one and are chosen using Importance Sampling [19][6], as given by Eq. (6):

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|x_t^i)\, p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{1:t-1}^i, z_t)} \quad (6)$$

Where $q(.)$ is a proposal importance density [6]. The described method, also known as the Sequential Importance Sampling algorithm, thus recursively propagates weights and particles as each measurement is received to the next time step. Finally, the state estimate of the target is obtained by Eq. (7)

$$\bar{x}_t = \sum_{i=1}^{N} x_t^i w_t^i \quad (7)$$

2.2 Sequential Importance Resampling

PF methods suffer from two inherent problems: sample degeneracy (particles distributed too widely) and impoverishment (particles being over concentrated) [6]. It has been observed that after a few iterations very few particles will have significant weight while all other particles will have negligible weight; thus these particles do not contribute their role in the representation of true state posterior and result in premature convergence [19]. The difference between impoverishment and degeneracy is that computing resource converges prematurely by weights in degeneracy problem and by particles in the impoverishment problem [19]. To overcome these problems, resampling [19]-[22] is used. Resampling replaces the degenerated particles with an equal number of new particles of equally-weighted (weights reset to $w_t^i = 1/N_s$) particles by replicating each particle according to its weight (refer to Figure 2), i.e., particles with larger weights are replicated while particles with lower weights are eliminated. It is observed that the resampling step could be applied when the particle diversity is lower than the effective sample size [6]. More details about resampling techniques can be found in [20][21].

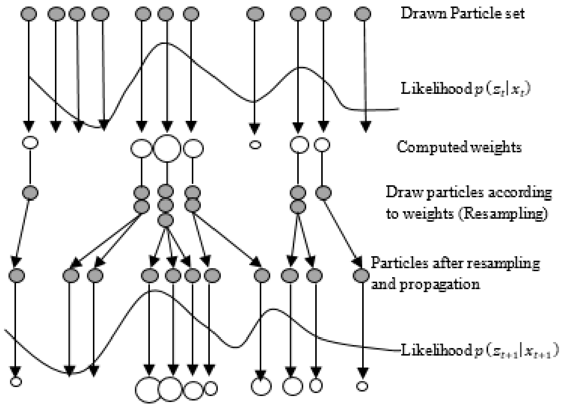


Figure2. Weight update and resampling

III. PARTICLE FILTER BASED VISUAL TRACKING

The basic steps of a generic Particle filter algorithm are described below:

Step 1(Initialization): Generate N samples $\{x_t^i\}_1^N$ from the prior distribution, $p(x_0)$

Step 2(Sampling): Draw prior particles $\{x_t^i\}_1^N$ from $p(x_t|x_{t-1}^i)$.

Step 3(Weighting): Assign weights to particles according to Eq. (6) and normalize weights as $w_t^i = w_t^i / \sum_{i=1}^N w_t^i$.

Step4(Resampling): Draw particles $\{x_t^i\}_1^N$ using resampling procedure and reset weight to $w_t^i = 1/N$.

Step 5(Estimation): Estimate the mean state as $\bar{x}_t = \frac{1}{N} / \sum_{i=1}^N x_t^i$, and repeat from step 2 to 5 for the next time instant.

Major research issues in PF-based visual tracking are importance sampling(IS), sample degeneracy/impoverishment, track management, parallel implementation, tracking under complex environments, such as occlusion, illumination, fast motion, appearance change, etc.

3.1 IMPORTANCE SAMPLING

Importance sampling is a Monte Carlo technique that approximates a posterior distribution using weighted samples drawn from a proposal function q(.), another distribution that resembles closely with the posterior distribution, as direct sampling from the posterior distribution is infeasible [23], [24]. Several approaches have been proposed to date to design robust IS methods [25]-[28]. One such approach is multiple importance sampling (MIS) that provides a more robust solution to the standard IS approach that uses a single proposal PDF. MIS uses a set of proposal PDFs to approximate target PDF with the aim of better and more efficient state estimates (Refer [25]for various MIS schemes). Elvira et al. [26] proposed a unified framework that includes many of the MIS techniques and how to sample from a mixture of proposal PDFs.

IS strategies are well known for exponential growth in the variance of the likelihood estimation. To reduce variance for small sample sizes at the cost of the increase in the computational complexity, Heretical MIS [27] divides proposal densities into disjoint subsets and the partitioning was done after drawing samples for better robustness. Another IS method that brings a large reduction in variance is Particle Efficient IS [28], which is based on non-standard resampling weights. It is important to note that while designing IS strategies, the aim should be the reduction in the computational complexity of the sampling process.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

3.2 SAMPLE DEGENERATION OR IMPOVERISHMENT

The sample impoverishment occurs when there is a significant reduction in the distinct number of particles, thus not representing the true state posterior leading to tracking failures. This problem can be alleviated by effective resampling. For various resampling techniques, refer to [29].

One of the effective strategies for resampling is to maintain particle diversity. To maintain particle diversity, a procedure called roughening [30] is used that improves particle diversity by adding a random noise (zero-mean random variable) to each resampled particle, i.e., the posterior particles are modified after the resampling step. Another approach that maintains particle diversity is the Gaussian distributed resampling method [31] that generates new particles based on a Gaussian distribution.

To combat sample impoverishment, evolutionary algorithms, such as the genetic algorithm(GA) can be used. GA uses two operators: crossover and mutation, which can be effectively used to maintain particle diversity to avoid sample impoverishment. Xiaowei et al. [32] used GA in the resampling process to generate new particles which better approximate the true state of the target. Rodriguez and Moreno [33] also included GA inside the resampling process by representing weights as fitness values to mitigate sample impoverishment.

3.3 SINGLE TARGET TRACKING

Before tracking, the target is first detected either manually or automatically using an object detection module. Once the target is detected, its features are extracted to model the target [34]. A target is modeled and tracked using various cues, such as colors, textures [32],[35],[36], motion [37], and edges [38]. As the appearance of the target may change with time, the target model is updated from time to time [36],[39]-[42].

Single-cue based Tracking

Color [43]-[48] is the most widely used feature for tracking as it can handle partial occlusion, scale variation, object deformation, and orientation. Color-based PF tracking, generally, describes the target model by creating a color histogram [36],[42],[43],[47]-[49] of the target. Using some similarity criterion of histograms, the target model is compared with the possible candidates in each frame. To track in a dynamic environment, the target model is updated with time. Sugandi et al. [42] tracked a moving target with its color histogram. They successfully tracked objects with both known and unknown initial positions by utilizing the appearance condition and variance of the particle's position. Li [48] reduced the computational cost by representing the color distribution of the target with a very small number of histogram bins as compared to the traditional color histogram. Lahraichi et al. [45]created a histogram of the foreground object byremoving the background pixels using backgroundsubtraction based on a mixture of Gaussian models for the effective representation of the target for tracking.

Generally, color-based PF trackers are sensitive to gradual, sudden, and strong illumination changes [41],[46],[50]. To handle illumination variations, Wang et al., [41] mapped the object from color feature space to local entropy space and constructed a color local entropy object observation model in HSV color space. The advantage of using the image local entropy is that it is insensitive to illumination variation and geometric distortion. The method is found effective in dealing with illumination and a small camera dithering. However, the method fails in sudden illumination changes. To track an object under strong and sudden illumination changes, Martínez-del-Rincon et al. [13] proposed a framework that tracks location, shape, and appearance using Rao-Blackwellised PF by dividing the state into image space parameters and the characteristic parameters. However, the method is computationally expensive. Mukhtar and Xia [46] used NTSC color format in color-based PF as NTSC color information remains unaffected by illumination variation.

To handle partial and full occlusions and target appearance and motion changes, Bimbo and Dini [51] split the state vector into static and dynamic parts and modeled the noise in the measurement equation. Zhang et al. [39] used a classified-patch kernel particle filter to track a target under severe occlusion. To track a fast-moving object having complex motions, Motion-Adaptive PF [52] provides a solution that could predict both the velocity and acceleration of the tracked object. For a weak and maneuvering target, Zhichaoet al.[53] separated the target existence variable from the target state to handle multiple motions.

Multiple-cue based Tracking

To handle a variety of environmental situations, generally, a single feature is insufficient for tracking. Therefore, to increase robustness, multiple features are combined. Table 1 presents various visual cues used for single target modeling and tracking.

A histogram is an efficient way to model a target but it is not a stable representation under a complex environment. Therefore, for a robust target model in a complex scene, authors [32], [36]-[38] [45], [54] proposed to use multiple cues as a single cue is insufficient; however, this increases the computational complexity and processing cost. Xiaoweiet al.[32] used color and texture features and accordingly assigned weights based on the feature's reliability. Using GA for resampling, the method is found to be robust against illumination variation, structural deformation, and occlusion. Dai and Liu [36], computed state estimation by fusing the marginal likelihood of color and texture features. The method is found to be robust against partial occlusion and the presence of confusing colors in the background. Bhat et al. [54] used color feature for target appearance and KAZE features to represent target structure. The use of KAZE features give superior performance for matching target in two frames Their multi-feature fusion scheme tracked object in various environmental situations, such as outdoor and underwater.

Color-based tracking is efficient in terms of computational cost but fails when foreground and background objects are of similar color. Further, extracted motion information from a complex environment sometimes contains complex background information leading to tracking errors. To handle such a situation, Wen et al. [37] combined color and motion information using contours to track an object in a complex environment. For better discrimination between the foreground and background, Sun et al.[47] mapped the object in a multi-space, which is a linear combination of RGB pixel values. The method is found to be effective in separating the target from the background and helps the accurate localization of the target.

Multiple features integration increases robustness, but at the cost of increased processing time as compared to single features.

Computational Efficiency

Achieving real-time performance, with a large number of particles (required to represent a true posterior PDF), is still a challenging issue. A solution to this end is to reduce the number of particles, but this may result in poor representation of the true posterior density of the target, leading to tracking failure. Liu et al. [55] proposed a higher-order PF tracking method. They selected the most suitable particles using K-means clustering which reduces the number of particles for tracking. To decrease the number of required particles, Mozhdehi and Medeiros [56] used PF with CNN and correlation filters to track in challenging environments, such as occlusions.

For the parallel implementation of PF, particle generation and weight update can be easily implemented in parallel[48],[57],[58]-[60], but the resampling step requires the cumulative sum of weights which cannot be easily parallelized. Efforts have been put to implement the parallelized version [59] of the resampling step using GPU. Murray et al. [59] implemented alternative resampling schemes, Metropolis and Rejection resamplers, that can be readily parallelized by eliminating the cumulative sum of weights. Li [48] proposed GPU based efficient measurement model by implementing a color cooccurrence histogram using integral images. Their results show that the tracking time increases insignificantly as the number of particles increases.

Many vision-based applications for embedded systems are becoming popular. The limitation of such systems is that they have low computing power and memory; therefore, developing efficient algorithms, for such systems, has attracted the researcher's attention. By utilizing GA in the resampling step using parallel hardware, Rodriguez and Moreno [33] implemented FPGA-based PF, targeting embedded systems. Using multi-core architecture, parallel implementation of the resampling step and likelihood calculation by utilizing the computing power of embedded systems are presented by Truong and Kim [49]. An accelerated PF method based on GPU, which can be applied to Internet of Things based applications, is presented by Kim et al. [58]. The authors achieved increased processing speed by simultaneously processing the state update process and weight computation process.

3.4 MULTIPLE TARGET TRACKING

In multiple target tracking, more than one object is tracked simultaneously. Due to several targets, MTT is more challenging. Major research issues in MTT are varying number of targets, occlusions with other targets that can sometimes have a similar appearance, lost tracks, track maintenance, target hijacking (causing PF to update its current state to the wrong object during the merging of two or more objects), track model drifting, misdetection, etc. Figure 3 shows a few MTT research issues. Due to the presence of multiple targets, received measurements might belong to any target so mapping between targets and corresponding measurements is necessary before state estimation. Thus applying STT models directly to MTT leads to wrong results and tracking failures.

Two approaches are popular for PF-based MTT: decentralized and centralized. In the decentralized approach, a unique instance of a PF for each target is used [35], [57],[60], [61], whereas, in the centralized approach, the states of the tracked targets are concatenated to form a single state vector [62]-[64] . A typical MTT consists of two components: A data association method – for assignments of measurements to targets, and a filtering method – for tracking each target.

Data association methods provide solutions for overcoming the inherent problem of measurement origin uncertainty associated with a multi-target environment [65]. Commonly used approaches for data association are a joint probabilistic data association filter (JPDAF) [63],[65]-[70],[73] and multiple hypothesis testing (MHT) [71],[72]. JPDAF computes the joint probabilistic score based on all possible assignments of measurements to targets. The major limitation of JPDAF is that it enumerates all possible joint association hypotheses to compute the marginalized probability. To reduce the number of these hypotheses, a technique called gating, is used that considers the measurements falling in a validation region. He et al.[73] discussed the trajectory optimization using JPDAF. JPDAF is typically used for the known number of well-separated and few targets. On the other hand, MHT provides a systematic solution to the data association problem by considering all valid data association hypotheses. It builds a tree of potential track hypotheses for each candidate target. As more measurements are received, the likelihood of each branch is computed and branches with lower probabilities are eliminated [72].

JPDAF becomes intractable for applications having a large number of targets and a cluttered environment (crowd surveillance where targets interact with each other). For handling a cluttered environment,Tchangoet al.[68] built an interaction graph, that evolves with time for target interaction.They used the means of particles to represent which targets are interacting with each other.

Table1: Various cues used for single target tracking

Reference	Tracked Object/ Dataset used	Visual Features	Model and Approach used	Number of Particles used	Target Model Update	Color Space	Tracking Performance
[51]	Toy car, Person	Color	Histogram	800	-	HSV	Tracks well when appearance and motion have fast changes and handles partial and full occlusion
[13]	Person's face	Color	Kernel density estimate	-	✓	RGB	Tracks target under strong illumination changes
[42]	Person	Color	Histogram	-	✓	RGB	Handles occlusion and tracks object with an unknown initial position
[37]	Person	Color, contour	Histogram	-	-	RGB	Fused multiple cues for efficient tracking under occlusion
[38]	CAVIAR	Color, Motion edge features	Histogram	-	-	-	Handles tracking in a complex environment, such as attitude change of a person and person having similar features
[48]	CAVIAR, Street Road sequences	Color	Cooccurrence histogram	320	-	RGB	Handles tracking in street sequences. Parallel implementation of measurement model on a GPU is attempted
[32]	Person's Face,	Color, Texture	Histogram, Local Binary	-	-	HSV	Tracks well when interference with similar

	person		Pattern(LBP) for texture				target occurs and under illumination variation and structural deformation
[41]	Car, Person	Color	Histogram	-	✓	HSV	Tracks well under illumination and occlusion
[46]	Person, football	Color	Histogram	150	-	YIQ	Tracks well under illumination changes, occlusion and moving background
[47]	Ship, Person	Color	Histogram	30	-	RGB	Discriminates foreground object from background efficiently
[36]	Car, Radio controlled helicopter	Color, Texture	Histogram, LBP	-	✓	-	Handles tracking in cluttered environment but fails when target in similar foreground and background color
[52]	OTB2013	Color	Histogram	50 to 300	-	HSV	Handles occlusion, background clutter, affine transformation and abrupt motion
[45]	PETS2012	Color, Gabour features	Histogram, Gabour features	-	-	HSV	Handles partial occlusion and discriminates foreground object from background well when both are of similar color
[49]	Person	Color	Histogram	300	-	RGB	Tracks target in complex color environment and handles occlusion efficiently
[88]	OTB2013	Color, Convolution features	Histogram	600	-	-	Handles local appearance changes and partial occlusion
[40]	OTB2015	Color Deep features	Histogram	40	✓	HSV	Fusion of deep features and handcrafted features is more effective for tracking under background clutter, illumination variation, fast motion and occlusion
[39]	OTB2013	Grey-scale features	Histogram	-	✓	-	Tracks well under occlusion, background clutter, illumination change, rotation, and scale change but lacks computational efficiency
[54]	OTB2015, CAVIAR, VOT16, ReefVid	Color, KAZE features	Histogram	50,100	-	HSV	Tracks well under occlusion, outdoor scenes and underwater environment.
[43]	OTB2013, OTB2015,	Color, Deep	Histogram	200, 600	-	RGB	Success ratio plots show good results under blurring

	LITIV	features					and occlusion
✓ - means target model is updated with time, - means not reported in the paper.							

To handle a large number of targets, Rezatofighi et al., [67] computed the JPDA score by reformulating the individual JPDA assignment score as a series of integer linear programs. They used m-best solutions to approximate the JPDA joint score.

Random Finite Set (RFS) [74]-[76] is a promising and alternative approach to JPDAF and MHT for avoiding the wrong association of measurements with targets. It performs MTT using the Bayesian state estimation approach. RFS extends the standard Bayes filter from STT to MTT problems by modeling the multi-target state and multi-target measurement by Finite Set Statistics (FISST), a tool for the information fusion, representation, and manipulation of RFS. In the RFS framework, the multi-target state is represented as a finite set of individual single target states. The predicted and updated posterior multi-object densities are then used to obtain the posterior PDF of multi-object states[74]. Ristic et al. [74] discuss various RFS-based filters in detail. The major limitation of RFS is that it is computationally demanding and practical for only a small number of targets.

JPDA and MHT work well when targets are limited in number, distinct, and far from each other. Also, these methods do not include labels, therefore these methods estimate the number of targets and their states without building connected tracks. On the other hand, label-based trackers [75]-[77] assign a unique label to each target state and link state estimate along time to form connected tracks. Papi and Kim [75] used a labeled RFS by assigning a unique label in the target state vector for track management. They used super-positional cardinalized probability hypothesis density(PHD) for multi-target sampling. PHD propagates the first-order moment of the multi-target posterior instead of the full posterior itself [78]. Wang et al.[77] used adaptive labeling for storing additional information, such as the size, width, and height of each tracked object.

Apart from the above methods, there have been other attempts for MTT. Li et al. [35] represented each target by a state particle set and used color and texture features in the measurement model of the PF framework. Data association is done by using an incidence matrix showing the correlation degree of observation values and particle set. Bai et al. [62] used PF over sets for MTT. For identifying each target from particles, they used the Expectation-Maximization method.

Due to many targets, the dimension of the state vector increases for MTT, leading to high computation time. Further with the data association step, computational complexity increases as the number of targets increases. Therefore, to achieve real-time performance with a large number of targets, authors [57],[64],[78],[79] explored parallel processing. The resampling step and weight update step are the two main challenging operations for parallel implementation as they require the joint processing of particles. Parallel resampling and weight update for the PHD filter on multiple processors are proposed in Coco and Cavallaro[78] and Li et al., [79]. For the parallel implementation of PF, Sutharsanand Kirubarajan[64] used dynamic scheduling and load balancing algorithms in a primary-secondary architecture. While achieving parallelization, allocation of particles among processors and reduction in communication overheads are to be considered. A summary of various PF-based MTT methods is shown in Table 2

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

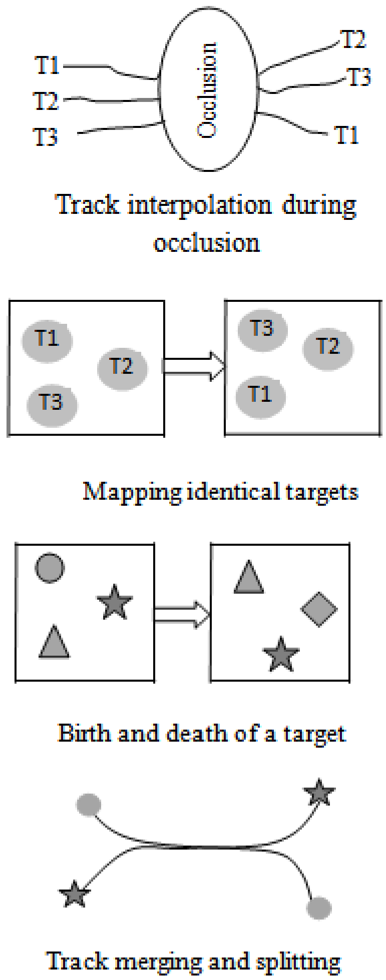


Figure 3: Some Multiple target tracking issues

3.5 DEEP LEARNING AND PARTICLE FILTER BASED TRACKING

The traditional PF-based visual trackers use hand-crafted measurement models which perform well in less complex scenarios, but for real-life complex scenarios, their performance is poor. For such situations, we require a more robust measurement model to distinguish a target in a complex background. In recent years, the convolutional neural network (CNN) has achieved great success in object detection and classification. Due to its high visual discriminative ability and automatic visual feature learning, researchers have reported some works on CNN-based measurement models in PF-based visual tracking. Further, with the availability of economical high-end GPU computers and many visual tracking datasets for training, deep learning, especially CNN, integrated with PF has proven to be more effective in target tracking, in terms of accuracy, than traditional methods.

For creating a CNN-based measurement model, we can either use a custom CNN model trained offline using a large dataset or use a generic pre-trained CNN and fine-tune the parameters for a given situation. In offline training, a large number of training data is used to train a network that automatically learns all generic features for tracking specific objects. In the second case, i.e., pre-trained CNN, standard CNN models are used after fine-tuning their parameters with the given situations. During tracking, a CNN-based measurement model is used to update the probabilities of the particles. A Block diagram of a CNN based particle filter is shown in Figure 4. Table 3 summarizes various visual tracking methods which use a combination of PF and deep learning method for visual tracking.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

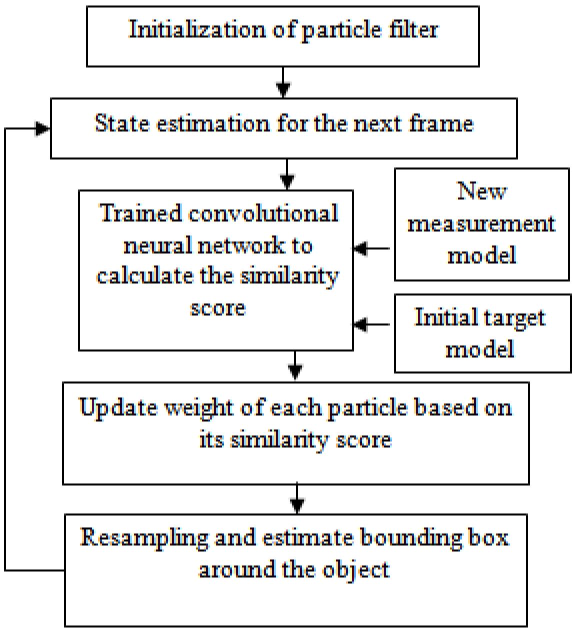


Figure 4: A block diagram of CNN based Particle Filter

Wang et al. [80] demonstrated an application of a combination of deep learning and PF for visual tracking. They used an offline-trained single-layer stacked denoisingautoencoder, a neural network, to generate image features that are more robust against variations, and used this as the measurement model to update the weights of the particles. Since online tracking involves the classification of image patches into the object and the background, they added a classification layer after the encode network. They also fine-tuned both the feature extractor and the classifier to adapt to the appearance changes of the moving object. They claimed that their tracker outperformed the state-of-the-art trackers on some benchmark videos, and also achieved real-time

performance due to low computational cost on simple GPU-based computers. However, due to the use of a small network with limited representation ability, the method sometimes produces lower accuracy when compared to traditional PF methods.

Gunawana et al. [81] used a combination of a stacked denoisingautoencoder and an extreme learning machine as the measurement model in a PF-based object tracker. The authors used a variant of autoencoder proposed by Wang et al. [80]; they used offline training on tiny images [82]. They reported that the running time of the tracking algorithm can be improved by using an extreme learning machine, and implemented the algorithm to track a single object only. Tyan and Kim [83] combined CNN and a particle filter by training a CNN model offline and used PF for tracking. They used a very compact CNN model to meet the real-time performance and updated the weights of every particle according to the CNN model. Their method outperformed state-of-the-art algorithms in handling object deformation, scale, and illumination changes, however, the method lost the target in case of fast motion and motion blurring.

The major limitation of deep learning and PF-based visual trackers is the training of the neural network model for general-purpose tracking. This requires a large amount of training data and also consumes lots of training time. To overcome these limitations, some authors [43],[84] have used pre-train networks already trained on large datasets. Youssef et al.[43] used AlexNet [86] which is pre-trained on the ImageNet database to classify 1000 objects of different classes, and integrated it in a PF framework to get more probable candidates around the region of interest. Quin et al. [40] also used a compact pre-trained network[86]. The output of the network is combined with the HSV color histogram to update the particle weights in a PF-based tracker.

To track multiple targets, i.e. MTT in a video, Xia et al. [44] used the mutual supervision between the convolutional neural networks-based multi-object detection and the particle filter-based tracking. They used features from different convolution layers of VGG-16 [87] to describe different views of the object. They reported that high-level features consisting of semantic information are good at distinguishing intra-class objects, whereas low-level features describing local

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

details of the target are good to distinguish between similar targets in the background. Based on the above findings, they used VGG-16 to identify the target regions using the feature maps output by Conv4-3 and Conv5-3. Once the regions of multiple objects are identified, they use the PF algorithm to track them. They further used the CNN model to correct the multi-object tracking trajectory generated by the tracker. This collaborative tracking helped them to track objects for a long time and achieves good accuracy. Their method can track rigid as well as a deformable object but need further improvement to objects with a large interference condition, such as low-light night vision and poor visibility scenes.

Chu et al. [88] designed the measurement model for their PF visual tracker by fusing the target color features and high-order features extracted through a convolution neural network. This combined appearance model is claimed to be more robust and accurate for tracking objects in a complex scene such as illumination, appearance change, and partial occlusion. They used a parameter to regulate the weights of the color model and the CNN model in the measurement model. This helped in improving the discriminative power of the measurement model. For example, when a target is partially occluded in a complex background, the global positioning advantage of the color model should be fully used. Similarly, when the color of the background is similar to the object, the CNN model should be given more weight. To make the tracking more robust, they updated the matching template in real-time. Experimentally, they found that the method has superior performance as compared to the normal PF-based tracker in a complex environment.

Another challenge in visual tracking is to update the target model which changes due to illumination variation, target deformation, scale change, cluttered background, etc. Therefore, the target model is to be updated as the initial target model may not be robust after few frames. To better discriminate the target from the background, Zhang et al. [89] employed local structure and geometric information among the targets in consecutive frames. Their measurement model was able to adapt to target appearance which alleviated target drifting and handles target deformation and background clutter efficiently. They used a two-layer CNN to learn representation for tracking. To model temporal changes in the target, Gurkan et al. [90] proposed the integration of a color particle filter and a deep learning-based object detector. After getting the bounding box of the target obtained using Mask R-CNN [91] as the object detector, the PF tracker generates all candidate bounding boxes, and simultaneously all candidate bounding boxes are also generated by Mask R-CNN. The tracked object is estimated by a decision scheme applied on bounding boxes obtained by Mask R-CNN and PF. Their method outperforms other methods in illumination and abrupt scale changes in a long-term tracking.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

Table 2: A summary of various PF based Multiple target tracking methods

Reference	Data Association Technique /Approach used	Datasets used	Features used	Remark
[60]	Decentralized approach	TUD crossing, PETS2009	Color	Proposed a state graph where each node represents the state of each track as partial, tracked, occluded, and lost which helps in linking tracks.
[72]	MHT	PETS2009, MOT challenge	-	Used online trained appearance model for each track of MHT.
[67]	JPDA	PETS2009, TUD Stadtmitte, MOT challenge	-	To get an accurate estimate of the complete JPDA, the authors reformulated the calculations of individual JPDA assignment scores and approximated the joint score by the m-best solutions.
[77]	Labeled RFS	PETS2009	-	Each target uses separate PF sets. Data association for maintaining tracks is done using the Hungarian algorithm.
[35]	State particle set	CAVIAR, PETS2012	Color, texture	Uses a correlation matrix to represent the correlation between object observation value and particle set of objects.
[62]	PF over sets	PETS2009	-	Used Expectation-maximization based target identification to separate each target from particles.
[57]	Decentralized approach	CAVIAR, ETH Central, TUD Crossing, PETS2009	-	Mapped MTT using PF to GPU and achieved five-fold speed up when compared to CPU for the same level of accuracy
[61]	Decentralized approach	PETS2009, TUD Stadtmitte	Color, edge direction, depth	Multiple features are integrated with PF to improve the tracking accuracy of each target and to make them adaptive to complex backgrounds.
[44]	Multi-Object regions generated using CNN	PETS2009	Color	Handles partial occlusion and disappearance of a target for a short duration by mutual supervision of CNN based multi-object detector and PF tracker
- means not reported in the paper.				

Table 3: Deep learning and PF based trackers

Reference	Datasets used	Attributes used for Evaluation,	Deep learning network used (pre-trained/proposed)	Number of Particles used	Tracking Performance

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

[80]	OTB2013	OTB2013 (SA)	Stacked denoisingautoencoder	1000	Detects appearance changes. Sometimes tracking leads to target drifting
[81]	OTB2013	Partial and full occlusion	An autoencoder	-	Tracks well under partial and full occlusion
[89]	OTB2013	OTB2013 (AA)	A two layer CNN	600	Handles target drifting, deformation, and background clutter well
[88]	OTB2015	OTB2015 (AA)	A two layer network	600	Handles local appearance changes and partial occlusion well
[40]	OTB2013	OTB2013 (AA)	A five layer network	40	A quantitative comparison shows that the tracker tracks well under fast motion, illumination variation, and deformation
[83]	OTB2015	OTB2015 (AA)	A five-layer CNN	-	Handles deformation, scale, and illumination changes but loses track in fast motion and motion blurring
[90]	VOT2015, VOT2016, VOT2017, OTB2015	VOT (SA)	Mask R-CNN	400	Abrupt appearance and scale changes are well handled in long-term tracking but fail under abrupt illumination changes and low resolution.
[44]	ETH Central, TUD crossing, PETS2012, Road traffic videos	PETS2012 (AA)	VGG-16 network	-	Handles targets disappearance and reappearance efficiently in long-term tracking of road traffic videos
[84]	OTB2013, OTB2015, UAV123, LaSOT	OTB(AA) and UAV123(AA)	VGG-M network	-	Outperforms under motion blur, in-plane rotation, fast motion, and out of view in OTB dataset. Performs well in long-term tracking.
[43]	OTB2015, LITIV	computed central location error and overlap ratio	AlexNet	100,200 and 600	Evaluated the tracker on 17 sequences and computed Central location error and overlap ratio
AA- All attributes of the dataset, SA-Some attributes of the dataset, - means not reported in the paper.					

IV. BENCHMARK DATASETS FOR VISUAL TARGET TRACKING

Looking at the wider applications of visual trackers, a large number of benchmark datasets for visual tracking have been developed. Most of the tracking datasets contain video sequences for short-term (ST) tracking, but in recent years, datasets for long-term (LT) tracking have been developed. In ST tracking, the trackers are not required to perform re-detection once the tracker is lost, but in LT tracking, the target disappears and re-appears in the scene, and hence trackers are required to perform re-detection once the target is lost; therefore, a target re-detector is required [92]. The major difference between ST and LT tracking is that in ST tracking, the target is always in view, whereas in LT tracking, the target may disappear from the view, and may re-enter in the view. LT is still a major challenge for visual-tracking researchers as this requires a huge database to train trackers [93].

In this section, popular object tracking benchmark datasets (refer Table 4), referred to in this paper, and their evaluation metrics are discussed in brief. These datasets are used by researchers to compare their tracking algorithms and to evaluate their performances. The datasets are described next:

Performance Evaluation of Tracking and Surveillance [94] (PETS2009 and PETS2012) - The latest tracking video sequences are provided under PETS2017, and the datasets for PETS2012 are the same as PETS2009 [95], targeting primarily surveillance applications. PETS 2009-S2 contains a dataset for people tracking with three types of the crowd: sparse, medium, and dense, and difficulty levels are L1, L2, L3. The activities included in the datasets are walking, running, and multiple flow merging. The task is to track all people in the sequence, and the results are reported as a 2D bounding box location for each individual. The dataset contains issues, such as occlusion, illumination changes, two persons with the same appearance, etc.

CAVIAR [96] (CAVIAR) – It contains two datasets with many video sequences for different scenarios, such as people walking alone, meeting with other people. The first dataset contains videos of the entrance lobby of INRIA Labs and the second one in a hallway in a shopping center. The first set contains several videos under six different scenarios: walking, browsing, resting, slumping or fainting, leaving bags behind, people/groups walking together and splitting up and two people fighting. The ground truth for each video sequence is provided in XML format, containing information, such as bounding boxes' locations and sizes, head and feet positions, etc. This dataset addresses the problems, such as occlusion, appearance changing, appearance, and disappearance.

Multiple Object Tracking Challenge Benchmark [97] (MOT)- The first release of the MOT challenge is in 2014. It provides a unified framework for multi-object tracking and contains annotated datasets and metrics for the evaluation of tracking methods. The website contains video sequences under MOT15, MOT16, MOT17, MOT20, and many more. Two sets of measures are used for evaluation: The CLEAR metrics (the measures used in the classification of events, activities, and relationships workshops), and the set of track quality measures. MOT15 contains video sequences of ETH Central, TUD Stadtmitte, and TUD crossing.

Online tracking Benchmark [16] (OTB2013 and OTB2015) **-** OTB2013 contains only 50 video sequences. OTB2015, the current version, is an extension of OTB2013. The benchmark contains 50 and 100 video sequences, each annotated by different attributes, There are eleven attributes considered in the datasets. They are: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, low resolution. The evaluation of trackers is based on two metrics: the accuracy of the tracker and location error. Success plots and precision plots are used for evaluating and comparing various tracker results [98],[99].

Visual Object Tracking Challenge [100] (VOT)- The first tracker challenge came as VOT2013 for STT. Since then, every year challenge is held. Datasets of VOT2016 [101], VOT2017[102], VOT 2018[103] have been referred to in this paper. It uses two metrics for evaluation. The first one is accuracy which computes the overlap ratio between the tracker and ground truth bounding boxes, and the second is robustness which is measured with respect to the frequency of tracking failure. Since VOT 2015, these metrics were replaced by the expected average overlap measure [92]. The current VOT2021 provides video sequences for ST and LT tracking and a platform for evaluation.

Large-scale single object tracking [104] (LaSOT) **-** It is a benchmark for large-scale single object tracking in long-term tracking. It contains video sequences where the target disappears and reappears again. Each frame in the sequences is provided with dense annotations. Each sequence is labeled with 14 attributes: illumination variation, full occlusion, partial occlusion, deformation, motion blur, fast motion, scale variation, camera motion, rotation, background clutter, low resolution, viewpoint change, out-of-view, and aspect ratio change. For evaluation, it uses three metrics: precision, normalized precision, and success plots [105].

A Benchmark and Simulator for UAV Tracking [106] (UAV123 and UAV20L) -It contains video sequences from an aerial viewpoint, a subset of which contains sequences for long-term aerial tracking. All sequences are fully annotated

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 3660-3680 www.ijaem.net ISSN: 2395-5252

with bounding boxes. Trackers are evaluated on twelve attributes [107].

LITIV [108] (LITIV**)**- The dataset consists of tracking people's heads in various conditions (occlusions, and many distractors, etc.). Video sequences are provided with ground truths containing centers, widths, and heights of tracked objects bounding boxes.

V. OPEN ISSUES FOR FURTHER RESEARCH

A robust PF target tracker should be able to track all targets efficiently in challenging environments. Many researchers have put their efforts to develop a robust PF tracker. However, the following are some issues and challenges that need to be addressed:

- The most important part of PF-based target tracking is to extract the most important and discriminating features of a target using particles. Further, integration of multiple features and selecting particles are of prime importance in PF-based tracking. Hence, selection and integration of features must be done efficiently, as an inappropriate representation may lead to tracking failures.
- PF-based trackers require a large number of particles to represent the probable locations of the target. A large number of particles improve tracking performance but increases computation costs resulting in a slow tracker. Research must be done to optimize the number of particles with a balance between accuracy and efficiency.
- To reduce computation cost, a parallel implementation of PF has been attempted by some researchers, but the resampling step requires a cumulative sum of weights, which cannot be easily parallelized. Efforts have been made in this direction with little success.
- The inclusion of computer vision tasks in IoT and embedded systems have attracted researcher's attention. Designing a PF-based tracker for such systems having low computational power and memory is a challenging research issue.
- The design of the measurement model for tracking objects with random birth and death of targets, handling partial and full occlusion of targets in complex environments are the important issues that need to be considered while designing a tracker.

Table 4: Datasets in recent years, referred in this paper, for visual tracking

Dataset	Year	Tracking Video Type	Frame Rate	Number of Attributes used for Evaluation	Number of Videos(year)
CAVIAR	2003	ST	25	-	28 and 44
OTB-50	2013	ST	30	11	50
LITIV	2014	ST	15	-	4
OTB-100	2015	ST	30	11	100
PETS	2009, 2012, 2016	ST	-	-	03 (2009, 2012), 14 (2016)
UAV123 UAV20L	2016	ST and LT	30	12	123 ST, 20 LT
LaSOT	2019	LT	30	14	1400
MOT	2014-2017, 2020	ST	7,10,14,25,30 (MOT 15) 14,25,30 (MOT16, MOT17) 25 (MOT20)	-	22 (2015), 14 (2016), 42 (2017), 8 (2020)
VOT	2013-2021	ST (2013-2017) ST and LT (2018- 2021)	30	-	16 (2013), 25 (2014), 60 (2015-2017), 60ST, 35LT (2018), 60ST, 50LT (2019-2020)
ST- Short term tracking video, LT- Long Term tracking video, - means not mentioned					

- MTT requires efficient data association techniques, track management, target labeling, target interaction, linking of tracks for reappearing targets, and addition of new targets with time. These issues must be addressed while designing an MTT method.
- PF integrated with deep learning has achieved significant improvements than traditional methods in challenging environments, such as illumination, partial occlusion, blurring, local appearance change, etc.
- Despite some success of deep-learning integrated with PF-based tracker in challenging environments, the tracker must be able to adapt to unseen environments. To achieve real-time performance, efficient online training of neural networks should be developed. Most deep learning-based trackers are trained for a specific domain. Further research is required to develop a generic domain-independent visual tracker.
- Most of the trackers have evaluated their results in short-term tracking videos. However, the evaluation of a generic tracker must be done on long-term tracking datasets as they represent a real-world scenario where targets disappear and reappear frequently because of occlusion.

VI. CONCLUSION

Visual tracking is one of the most widely researched topics in computer vision due to its many practical applications. Many techniques have been developed for visual tracking. Particle filter-based visual tracking has been widely researched due to its ability to handle real-life complex dynamic environments, random motions, multiple objects, and non-Gaussian noises of sensors. Further, the particle filter has been integrated with the convolutional neural network to produce the state-of-the-art technique for visual tracking. This paper reviews the recent advances in PF-based visual tracking. Moreover, an attempt has been made to explore and compare various advances of PF-based trackers in STT, MTT, and deep learning integrated PF trackers. Starting from traditional to deep learning-based PF trackers, we covered various aspects of PF, such as importance sampling and sample degeneration, designing PF-based trackers for a single target and multiple targets, and integration of PF and deep learning. Various trackers, their performances in challenging environments, and datasets used by these methods are also reported. The paper also reports the unresolved research issues to be further researched to handle challenges, such as target shape changes, illumination variations, occlusions, and cluttered and dynamic environments.

REFERENCES

[1] Abbass, M. Y; Kwon, K. C.; Kim, N.; Abdelwahab, S. A.; El-Samie, F. E. A., Khalaf, A. A. M., 2020“A survey on online learning for visual tracking”, The Visual Computer.

[2] Soleimanitaleb, Z.;Keyvanrad, M. A.;Jafari, A., 2019, “Object Tracking Methods: A Review”, International Conference on Computer and Knowledge Engineering, pp. 282-288.

[3] Marvasti-Zadeh, S. M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S., 2021,“Deep Learning for Visual Tracking: A Comprehensive Survey”, IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3046478.

[4] Elfring, J.; Torta, E.; Molengraft, R. V., 2021“ Particle Filters: A Hands-On Tutorial”, Sensors 2021, vol. 21, 438, 2021

[5] Godsill, S., 2019, “Particle Filtering: the First 25 Years and beyond”. IEEE International Conference on Acoustics, Speech and Signal Processing, UK, pp. 7760-7764.

[6] Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp,T., 2002 “ A Tutorial on Particle Filters for Online Nonlinear/Non Gaussian Bayesian Tracking” . IEEE Transactions on Signal Processing, vol. 50(2), pp. 174-188.

[7] Smal, I.; Meijering, E.; Draegestein, K.; Galjart, N.; Grigoriev,I.; Akhmanova, A.; Royen, M.E.; Houtsmuller, A. B.; Niessen, W., 2008, “Multiple Object Tracking in Molecular Bioimaging by Rao-Blackwellized Marginal Particle Filtering”, Medical Image Analysis, vol.12 (6), pp.764-777.

[8] Varas, D.; Marques, F., 2014,“Region-based particle filter for video object segmentation”, IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 3470-3477.

[9] Azzabou, N.; Paragios, N.; Guichard, F., 2010, “Image Reconstruction Using Particle Filters and Multiple Hypotheses Testing”, IEEE Transactions on Image Processing, vol. 19(5), pp.1181-1190, 2010.

[10] Isard, M.; Blake, A., 1998, “CONDENSATION—Conditional Density Propagation for Visual Tracking”, International Journal of Computer Vision, vol. 29(1), pp.5-28, 1998.

[11] Zhao, Y., 2018, “Design and Implementation of Vehicle Tracking System Based on Depth

Learning", International Conference on Advances in Image Processing, USA, pp. 138–143.
[12] Hu, J.; Lu, J.; Tan, Y.,2016, "Deep Metric Learning for Visual Tracking", IEEE Transactions on Circuits and Systems for Video Technology, vol. 26(11), 2056-2068.
[13] Martínez-del-Rincon, J.; Orrite, C.; Medrano, C., 2011, "Rao-Blackwellised Particle Filter for Colour-Based Tracking" , Pattern Recognition Letters , vol. 32, pp. 210-220.
[14] Ho, M. F.; Tseng, C.; Lien, C.; Huang, C., 2011"A multi-view vision-based hand motion capturing system", Pattern Recognition, vol. 44 (2), pp. 443-453.
[15] Eder, M.; Reip, M.; Steinbauer, G., 2021"Creating a robot localization monitor using particle filter and machine learning approaches" ApplIntell, https://doi.org/10.1007/s10489-020-02157-6.
[16] OTB 2013 and OTB 2015. URL: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html
[17] Rao, G.M.; Satyanarayana, C.; 2013 "Visual Object Target Tracking using Particle Filter: A Survey". I.J. Image, Graphics and Signal Processing, vol. 6, pp. 57-71.
[18] Wang, X.; Li, T.; Sun, S.;Corchado, J. M., 2017, "A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking" Sensors,vol. 17, 2707.
[19] Li, T.;Sattar, T. P.; Sun, S., 2012 "Deterministic Resampling: Unbiased Sampling to Avoid Sample Impoverishment in Particle Filters", Signal Processing, vol. 92, pp.1637-1645.
[20] Li, T.; Bolic, M.; Djuric, P. M., 2015, "Resampling Methods for Particle Filtering" Signal Processing Magazine , 70.
[21] Li, T.; Villarrubia, G.; Sun, S.; Corchado, J. M.; Bajo, J., 2015, "Resampling Methods for Particle Filtering: Identical Distribution, A New Method and Comparable Study" Frontiers of Information Technology & Electronic Engineering, vol.16(11), pp. 969-984.
[22] Fu,X.;,Jia, Y., 2010, "An Improvement on Resampling Algorithm of Particle Filters", IEEE Transactions on Signal Processing, vol. 58(10), pp. 5414-5420.
[23] Martino,L.; Elvira, V.; Camps-Valls, G., 2018, "Group Importance Sampling for Particle Filtering and MCMC", arXiv:1704.027714[stat.CO].
[24] Bugallo, M. F.; Martino, L.; Corander, J.,2015, "Adaptive Importance Sampling in Signal Processing" Digital Signal Process, vol. 47, pp. 36-49.
[25] Sbert, M.; Havran, V.; Szirmay-Kalos, L.; 2018 "Multiple Importance Sampling Revisited: Breaking the Bounds", EURASIP Journal on Advances in Signal Processing, 15.
[26] Elvira, V.; Martino, L.; Luengo, D.; Bugallo, M. F., 2017 " Generalized Multiple Importance Sampling", arXiv:1511.03095[stat.CO].
[27] Elvira, V.; Martino, L.; Luengo, D.; Bugallo, M. F., 2016 "Heretical Multiple Importance Sampling", IEEE Signal Processing Letters, vol. 23, pp. 1474-1478.
[28] Scharth, M., Kohn, R., 2013, "Particle Efficient Importance Sampling", arXiv:1309.6745v1 [stat.CO],(2013).
[29] Li, T.; Suna, S; Sattar, T. P.; Corchado, J. M., 2014, "Fight Sample Degeneracy and Impoverishment in Particle Filters: A Review of Intelligent Approaches", Expert Systems with Applications, vol. 41, pp. 3944-3954.
[30] Li, T.; Sattar, T. P.; Han, Q.; Sun, S., 2013, "Roughening Methods to Prevent Sample Impoverishment in the Particle PHD Filter". International Conference on Information Fusion, pp. 17-22.
[31] Choi, H. D.; Pak, J. M.; Lim, M. T.; Song, M. K., 2015, "A Gaussian Distributed Resampling Algorithm for Mitigation of Sample Impoverishment in Particle Filters". International Journal of Control, Automation, and Systems vol. 13(4), pp. 1-5.
[32] Xiaowei, Z.; Hong, L.;Xiaohong, S., 2013, "Object Tracking with an Evolutionary Particle Filter Based on Self-Adaptive Multi-Features Fusion", International Journal of Advanced Robotic Systems, 10,61, 2013.
[33] Rodriguez, A.; Moreno, F., 2015, "Evolutionary Computing and Particle Filtering: A Hardware-Based Motion Estimation System", IEEE Transactions on Computers, vol. 64(11), pp. 3140-3152, 2015.
[34] Dalal, N.;Triggs, B., 2005 "Histograms of Ooriented Gradients for Human Detection", CVPR, USA, 1, pp. 886–893, 2005.
[35] Li, H.; Liu, Y.; Wang, C.; Zhang, S.; Cui, X., 2016, "Tracking Algorithm of Multiple Pedestrians Based on Particle Filters in Video Sequences", Computational Intelligence and Neuroscience, 8163878, 2016.
[36] Dai, Y.; Liu, B, 2015, "Robust Video Object Tracking Using Particle Filter with Likelihood Based Feature Fusion and

Adaptive Template Updating", arXiv:1509.08182v1[cs.CV], 2015.
[37] Wen, Z.; Peng, Z.; Deng, X.; Li, S., 2011. Particle Filter Object Tracking Based on Multiple Cues Fusion, Procedia Engineering, 15, 1461-1465.
[38] Cao, J.; Li, W.; Wu, D., 2012, "Multi-feature Fusion Tracking Based on A New Particle Filter", Journal of Computers, vol. 7(12).
[39] Zhang, G.; Yang, J.; Wang, W.; Hu, Y. H.; Liu, J., 2019"Adaptive visual target tracking algorithm based on classified-patch kernel particle filter", EURASIP Journal on Image and Video Processing, 2019:20, 2019.
[40] Qian, X.; Han, L.; Wang, Y.; Ding, M., 2018,"Deep Learning Assisted Robust Visual Tracking With Adaptive Particle Filtering" , Signal Processing: Image Communication, vol. 60, pp. 183-192.
[41] Wang, H.; Wang, Q.; Li, Y.; Dai, Y., 2013,"Particle Filter Algorithm for Object Tracking Based on Color Local Entropy" , Advances in Mechanical Engineering, 961019.
[42] Sugandi, B.; Kim, H.; Tan, J. K.; Ishikawa, S., 2011, "Object Tracking Based on Color Information Employing Particle Filter Algorithm", Object Tracking, InTech, doi: 10.5772/15277.
[43] Youssef, A.; Aouatif, A.; Hanaa, H.; 2020"Object Tracking with Adaptive Particle Filter Tracker using Convolutional Neural Network", International Conference on Industrial Engineering and Operations Management, USA, pp.3872-3882.
[44] Xia, Y.; Qu, S.; Goudos, S.; Bai, Y.; Wan, S.,2019, "Multi-object tracking by mutual supervision of CNN and particle filter, Personal and Ubiquitous Computing", https://doi.org/10.1007/s00779-019-01278-1.
[45] Lahraichi, M.; Housni, K.; Mbarki, S.; 2017, "Particle Filter Object Tracking Based on Color Histogram and Gabor Filter Magnitude", International Conference on Big Data, Cloud and Applications, 78, pp.1-5.
[46] Mukhtar, A.; Xia, L., 2014, "Target Tracking Using Color Based Particle Filter", International Conference on Intelligent and Advanced Systems, Malaysia, 1-6.
[47] Sun, X.; Yao, H.; Lu, X.; 2014 "Dynamic Multi-Cue Tracking using Particle Filter", Signal, Image and Video Processing, vol. 8, pp. 95-101.
[48] Li, P., 2012, "An Efficient Particle Filter-based Tracking Method Using Graphics Processing Unit", J. Sign. Process. Syst. 68, 317-332.
[49] Truong, M. T. N.; Kim, S., 2017, "Parallel Implementation of Color‐Based Particle Filter for Object Tracking in Embedded Systems" Human-Centric Computing and Information Sciences ,7, 2.
[50] Chan, Y.M.; Huang, S.S.; Fu, L.C.; Hsiao, P.Y.; Lo, M.F., 2012, "Vehicle Detection and Tracking Under Various Lighting Conditions Using a Particle Filter", IET Intelligent Transport Systems, vol. 6(1), pp. 1-8.
[51] Bimbo, A. D.; Dini, F., 2011, "Particle Filter-Based Visual Tracking with a First Order Dynamic Model and Uncertainty Adaptation", Computer Vision and Image Understanding, vol. 115(6), pp. 771-786.
[52] Cao, S.; Wang, X.; Xiang, K., 2017, "Visual Object Tracking Based on Motion-Adaptive Particle Filter under Complex Dynamics", EURASIP Journal on Image and Video Processing, 76.
[53] Zhichao, B.; Qiuxi, J.; Fang, zheng, L., 2020, "Multiple Model Efficient Particle Filter Based Track-Before-Detect for Maneuvering Weak Targets", Journal of Systems Engineering and Electronics, vol. 31(4), pp. 647- 656.
[54] Bhat, P. G.; Subudhi, B. N.; Veerakumar, T.; Laxmi, V.; Gaur, M. S., 2020, "Multi-Feature Fusion in Particle Filter Framework for Visual Tracking", IEEE Sensors Journal, vol. 20(5), pp. 2405-2415.
[55] Liu, J.; Chen, Y.; Zhou, L.; Zhao, L., 2018,"Visual tracking in high-dimensional particle Filter", PLOS ONE 13(8): e0201872.
[56] Mozhdehi, R. J.; Medeiros, H., 2020"Deep Convolutional Likelihood Particle Filter for Visual Tracking", arXiv:2006.06746v1 [cs.CV].
[57] Jecmen, P.; Lerasle, F.; Mekonnen, A. A., 2017, "Trade-Off Between GPGPU Based Implementations of Multi Object Tracking Particle Filter", International Conference on Computer Vision Theory and Applications, Portugal. 10p.
[58] Kim, S.; Cho, J.; Park, D., 2017 "Moving-Target Position Estimation Using GPU-Based Particle Filter for IoT Sensing Applications", Applied Sciences,7,1152.
[59] Murray, L. M.; Lee, A.; Jacob, P. E., 2016, "Parallel Resampling in the Particle Filter", Journal of Computational and Graphical Statistics, vol. 25(3).

[60] Geronimo D.G., Lerasle, F., Pena , A. M. L., 2012, "State-Driven Particle Filter for Multi-Person Tracking", LNCS, 7517.
[61] Ma, Y.; Liu, Y.; Liu, S.; Zhang, Z., 2017, "Multiple Object Detection and Tracking in Complex Background", International Journal of Pattern Recognition and Artificial Intelligence, vol. 31(02), 1755003.
[62] Bai, A.; Simmons, R.; Veloso, M., 2017, "Multi-Object Tracking and Identification via Particle Filtering over Sets", International Conference on Information Fusion, China, pp. 1263-1270.
[63] Liu, J.; Han, C. Z.; Han, F.; Hu, Y., 2012, "Multiple Maneuvering Target Tracking by Improved Particle Filter Based on Multiscan JPDA", Mathematical Problems in Engineering, 372161.
[64] Sutharsan, S.; Kirubarajan, T., 2012, "An Optimization-Based Parallel Particle Filter for Multitarget Tracking", IEEE Transactions on Aerospace and Electronic Systems vol. 48(2).
[65] Jinana, R., Raveendran, T., 2016, "Particle Filters for Multiple Target Tracking", International Conference on Emerging Trends in Engineering, Science and Technology, Procedia Technology, 24, pp. 980-987.
[66] Ekman, M., 2008, "Particle filters and data association for multi-target tracking," International Conference on Information Fusion, pp. 1-8.
[67] Rezatofighi, S. H.; Milan, A.; Zhang, Z.; Shi, Q.; Dick, A.; Reid, I.,2015"Joint Probabilistic Data Association Revisited", IEEE International Conference on Computer Vision, 3047-3055.
[68] Tchango, A. F.; Thomas, V; Buffet, O.; Dutech, A.; Flacher, F., 2014,"Tracking Multiple Interacting Targets Using a Joint Probabilistic Data Association Filter", Int. Conf. on Information Fusion, pp. 1-8.
[69] Yang, T.; Huang, G.; Mehta, P.G., 2013, "Joint Probabilistic Data Association-Feedback Particle Filter for Multiple Target Tracking Applications", arXiv:1303.1214v1 [math.NA].
[70] Jaward, M.; Mihaylova, L.; Canagarajah, N.; Bull, D., 2006 "Multiple Object Tracking using Particle Filters, IEEE Aerospace Conference, USA, 8.
[71] Zhang, Y.; Yang, Y.; Wei, S.; Wang, J., 2016, "Fast Data Association Approaches for Multi-target Tracking", CIE International Conference on Radar, pp.1-5.
[72] Kim, C.; Li, F.; Ciptadi, A.; Rehg, J. M., 2015, "Multiple Hypothesis Tracking Revisited", ICCV, pp. 4696-4704.
[73] He, S.; Shin, H.; Tsourdos, A., 2020, "Trajectory Optimization for Multitarget Tracking Using Joint Probabilistic Data Association Filter", Journal of Guidance, Control, And Dynamics, vol. 43(1).
[74] Ristica, B.; Beard, M.; Fantacci, C., 2016, "An Overview of Particle Methods for Random Finite Set Models", arXiv:1602.03945v1[cs.SY].
[75] Papi, F.; Kim, D. Y., 2015"A Particle Multi-Target Tracker for Superpositional Measurements using Labelled Random Finite Sets", IEEE Transactions On Signal Processing, vol. 63(16).
[76] Garcia-Fernandez, A. F.; Grajal, J.; Morelande, M. R., 2013"Two-Layer Particle Filter for Multiple Target Detection and Tracking", IEEE Transactions on Aerospace and Electronic Systems, vol. 49(3), pp. 1569-1588.
[77] Wang, Z.; Xu, B.; Huang, F., 2015"Real-Time Multiple Object Tracking in Particle Filtering Framework using Codebook Model and Adaptive Labeling", International Conference on Research in Adaptive and Convergent Systems, 141-145.
[78] Coco, M. D.; Cavallaro, A., 2014, "Parallel Particle-PHD Filter", International Conference on Acoustics, Speech and Signal Processing, pp. 6578-6582.
[79] Li, T.; Sun, S.; Bolić, M.; Corchado, J. M., 2016, "Algorithm Design for Parallel Implementation of The SMC-PHD Filter", Signal Processing, 119115-127.
[80] Wang, N.; Yeung, D. Y., 2013, "Learning A Deep Compact Image Representation for Visual Tracking". Proceedings, Advances in Neural Information Processing Systems, Tahoe, pp. 809-817.
[81] Gunawana, A.A.S.; Fanany, M. I.; Jatmiko, W., 2014, "Deep Extreme Tracker Based on Bootstrap Particle Filter", Journal of Theoretical and Applied Information Technology, vol. 66 (3), pp. 857-863.
[82] Torralba, A.; Fergus, R.; Freeman, W., 2008, "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30(11), pp. 1958-1970.
[83] Tyan V.; Kim, D., 2018"Convolutional Neural Network with Particle Filter Approach for Visual Tracking", KSII Transactions on

Internet and Information systems, vol. 12(2), pp. 693-709.
[84] Tang, Y.; Liu, Y.; Huang, H.; Liu, J.; Xie, W., 2020, "A Scale-Adaptive Particle Filter Tracking Algorithm Based on Offline Trained Multi-Domain Deep Network", IEEE Access, 8, 31970-31982.
[85] Fan, H.; Ling, H., 2017, "SANet: Structure-aware network for visual tracking", IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, USA, 2217-2224.
[86] Krizhevsky, A.; Sutskever, I.; Hinton, G.E., 2012, "Imagenet Classification with Deep Convolutional Neural Networks", International Conference on Neural Information Processing Systems, pp. 1097–1105.
[87] Simonyan, K.; Zisserman, A.,2015,"Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556v6, 2015.
[88] Chu, H.; Wang, K.; Xing, X., 2018, "Target Tracking via Particle Filter and Convolutional Network", Journal of Electrical and Computer Engineering, 5381962.
[89] K. Zhang, Q. Liu, Y. Wu, M. Yang, "Robust Visual Tracking via Convolutional Networks Without Training", IEEE Transactions on Image Processing, vol. 25(4), pp. 1779-1792, 2016.
[90] Zhang, K.; Liu, Q.; Wu, Y.; Yang, M., 2016, "Robust Object Tracking via Integration of Particle Filtering with Deep Detection", Digital Signal Processing, vol. 8, pp. 112-124.
[91] Akok, B.; Gurkan, F.; Kapla, O.; Gunsel, B., 2017, "Robust Object Tracking by Interleaving Variable Rate Color Particle Filtering and Deep Learning", IEEE International Conference on Image Processing, Beijing, pp. 3665-3669.
[92] Kristan, M. et al., 2019, "The Seventh Visual Object Tracking VOT2019 Challenge Results", IEEE/CVF International Conference on Computer Vision Workshop, Korea (South), pp. 2206-2241.
[93] Wang, J.; Yang, H.; Xu, N. et al., 2021,"Long-Term Target Tracking Combined with Re-Detection". EURASIP Journal on Advances in Signal Processing, 2.
[94] http://www.cvg.reading.ac.uk/PETS2012/a.html
[95] Ferryman, J.; Shahrokni, A., 2009, "PETS2009: Dataset and challenge", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, USA, 1-6.
[96] https://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
[97] MOT, URL: https://motchallenge.net/
[98] Wu, Y.; Lim, J.; Yang, M., 2013, "Online Object Tracking: A Benchmark", IEEE Conf. Computer Vision Pattern Recognition, USA, pp. 2411-2418.
[99] Wu, Y.; Lim, J.; Yang, M., 2015, "Object Tracking Benchmark", IEEE Trans. Pattern Anal. Mach. Intell., vol. 37(9), pp.1834 - 1848.
[100] https://www.votchallenge.net/challenges.html
[101] Kristan et al. M., 2016, "The Visual Object Tracking Vot2016 Challenge Results", The European Conference on Computer Vision Workshops, pp. 777-823,
[102] Kristan et al. M., 2017, "The Visual Object Tracking Vot2017 Challenge Results", The IEEE International Conference on Computer Vision Workshops, 1949-1972.
[103] Kristan et al. M., 2018, "The Sixth Visual Object Tracking Vot2018 Challenge Results", The European Conference on Computer Vision Workshops, Lecture Notes in Computer Science, 11129.
[104] http://vision.cs.stonybrook.edu/~lasot/download.html
[105] Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H., 2019 "LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking", IEEE/CVF Conference on Computer Vision and Pattern Recognition, USA, 5369-5378.
[106] https://cemse.kaust.edu.sa/ivul/uav123
[107] Mueller, M.; Smith, N.; Ghanem,B., 2016, "A Benchmark and Simulator for UAV Tracking", European Conference on Computer Vision, The Netherland, pp. 445-461.
[108] URL:http://www.polymtl.ca/litiv/en/codes-and-data