Target Tracking using a Mean-Shift Occlusion Aware Particle Filter

Pranab Gajanan Bhat, Badri Narayan Subudhi, *Senior Member, IEEE*, T.Veerakumar, Gaetano Di Caterina and John J. Soraghan, *Senior Member, IEEE*

***Abstract*—Most of the sequential importance resampling tracking algorithms use arbitrarily high number of particles to achieve better performance, with consequently huge computational costs. This article aims to address the problem of occlusion which arises in visual tracking, using fewer number of particles. To this extent, the mean-shift algorithm is incorporated in the probabilistic filtering framework which allows the smaller particle set to maintain multiple modes of the state probability density function. Occlusion is detected based on correlation coefficient between the reference target and the candidate at filtered location. If occlusion is detected, the transition model for particles is switched to a random walk model which enables gradual outward spread of particles in a larger area. This enhances the probability of recapturing the target post-occlusion, even when it has changed its normal course of motion while being occluded. The likelihood model of the target is built using the combination of both color distribution model and edge orientation histogram features, which represent the target appearance and the target structure, respectively. The algorithm is evaluated on three benchmark computer vision datasets: $OTB100$, $VOT18$ and $TrackingNet$. The performance is compared with fourteen state-of-the-art tracking algorithms. From the quantitative and qualitative results, it is observed that the proposed scheme works in real-time and also performs significantly better than state-of-the-arts for sequences involving challenges of occlusion and fast motions.**

***Index Terms*—Object tracking, Occlusion, Mean-Shift, Particle filter.**

I. Introduction

Visual Tracking is among the most important and challenging research areas in computer vision [1]. The aim of tracking is to estimate the motion of a target as it moves around the scene in the image plane [2]. Motion estimates are generally derived by using predictors such as linear regression techniques, Kalman Filter and Particle Filter. The particle filter works on the principle of sequential importance resampling and can be used as a motion estimator when the target has non-Gaussian state space. Many methods have been proposed in literature to overcome the limitations of particle filter, namely degeneracy problem, sample impoverishment and determining the exact number of particles for tracking. Most of the algorithms arbitrarily select larger number of particles to achieve appreciable results. Although results achieved are satisfactory, the number of computations increase manifold for each new particle considered. To tackle this problem and allow tracking using a smaller set of quality particles, integration of Mean-Shift and particle filter to fuse the advantages of both in visual tracking was proposed by Maggio et al. [3]. Shan et al. [4] proposed integration of mean-shift algorithm along with the particle filter (MSEPF) to track the movement of a hand. Here, the modes of distribution of particles are detected using the mean-shift search, and are weighed based on their likelihood model. Since modes correspond to high likelihoods, only the significant particles in the proximity of these modes are selectively resampled. This integration of mean-shift and particle filter leads to tracking with a fewer number of particles as compared to the case where mean-shift is not used. The integration also enables significant reduction in processing time due to consideration of a smaller particle set.

One critical factor which makes tracking more challenging is the variations in object appearance. These variations are caused by several factors such as occlusion, illumination change, fast motion, pose variation and noise disturbance [5]. Among all these challenges, this article focuses on the challenge of occlusion. Occlusion is said to have occurred when the attributes of the tracked target are unavailable, even when the target is present in the scene. For partial occlusions, designing the motion model is relatively easy due to partial visibility of the target as compared to full occlusion case, where the target is not visible. If the target changes its speed or direction of motion or both when it is fully occluded, developing the motion model is even more challenging.

In this article, a Mean-Shift Occlusion Aware Particle Filter (MSOAPF) is proposed to track the targets using both HSV color and edge oriented histogram (EOH) features of the target. The particle filter is integrated with the mean-shift algorithm so as to reduce the number of particles required to maintain state hypotheses. The mean-shift algorithm allows to focus only on the particles with high likelihood, thus making the resampling process more efficient. The weights on particles are dependent on the likelihood of the color and EOH features of the reference target and probable candidates, which is computed using the Bhattacharya coefficient. Occlusion is detected based on the correlation coefficient between the target and filtered candidate and, if detected, fast failure recovery from occlusion is achieved by switching the motion model to random walk model from the constant velocity model. The switching of motion model results in gradual increase in the

P. G. Bhat and T. Veerakumar are with the Department of Electronics and Communication Engineering, National Institute of Technology Goa, Ponda, Goa, 403401 India. (e-mail: pannubhat@gmail.com; tveerakumar@yahoo.co.in)

B. N. Subudhi is with Department of Electrical Engineering, Indian Institute of Technology Jammu, Nagrota, 181 221 India (e-mail: subudhi.badri@gmail.com)

G. Di Caterina and J.J. Soraghan are with the Centre for Signal and Image Processing, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, U.K. (e-mail: gaetano.dicaterina@strath.ac.uk; j.soraghan@strath.ac.uk)

spatial spread of particles allowing more opportunities for quick recapturing of the target post-occlusion. The algorithm is capable of recapturing the target even if it changes its direction of motion while under occlusion. This is a fresh attempt to develop an iterative feature-based tracker which can track the target not only with a very smaller number of particles, but with an improved occlusion handling ability using integration of particle filter framework, mean-shift algorithm and correlation-based occlusion detector using switching of motion models. In summary, the proposed method has two fold contributions: integration of three methodologies for efficiently tracking the targets with fewer number of particles and switching of motion model of target to quickly recapture the target in complex scenes. Results illustrate that the MSOAPF performs better than some of the advanced learning-based methods, for the challenges of occlusion and fast-motions and gives comparable results in terms of tracking accuracy for other challenges. Further, the proposed method is computationally very fast and operates at a rate of 38.2 frames per second making it suitable for real-time applications.

The organization of this article is as follows. Various state-of-the-art tracking algorithms are presented in Section II. Section III gives description about the particle filter framework and mean-shift algorithms used for tracking. The proposed method is elucidated along with the help of a flowchart in Section IV. In Section V, a brief of experimental setup and datasets used are presented, while the results and analysis of the proposed scheme are carried out in Section VI. Finally, the conclusions of the work are drawn in Section VII.

II. Related Works

Various methods exist in literature to recover the target after occlusion. Meshgi et al. [6] proposed a particle filter algorithm to handle complex occlusion scenarios, which prevented loss of target by predicting emerging occlusions and also achieving quick occlusion recovery of the target. Duan et al. [7] presented a method for detection and recovery from occlusion while tracking a target using particle filter. Line et al. [8] used a patch-based appearance model for handling occlusions wherein, when the weighing based on the color model is less than a particular threshold, the particles are weighed based on Speeded Up Robust Features (SURF). Overall, it is observed that most of the algorithms opt for template matching variations to recapture the target from occlusion wherein entire image area is searched to locate the target in the scene. Some algorithms like [8] use complex feature set to recapture the target. In the proposed algorithm, instead of opting for template matching or its variations, the motion model is tweaked so as to incorporate all the probable target reoccurring points. This allows for faster recapturing of the target without having to search for the entire image area and also without using any complex feature set for recovery. The proposed approach also handles the changes in motion and direction of motion of the target when it is fully occluded.

Recently, there has been a shift from conventional optical flow method to feature-based visual tracking in tracking applications. The extraction of features for object representation has to be done in such a way that the target is uniquely distinguished from other objects in the feature space [2]. Color model is one of the most widely used feature for tracking as it is robust to partial occlusions, scale variance and rotation. Nummiaro et al. [9] used color histogram in particle filter framework for tracking a target. The approach mainly involves adapting the target model during stable image observations for making the tracker immune to changes in illumination. However, a single feature is not sufficient to describe the target completely and hence a host of cues have to be fused in a principled manner to increase the tracking reliability. Brasnett el al. [10] designed a particle filter for tracking using multiple cues (edge, color and texture). A similar method for target tracking was developed by Niu et al. [11], where fusion of color and SURF is incorporated in the particle filter framework. Rahimi et al. [12] proposed a particle filter tracking method using color, cellular local binary pattern and Histogram of Oriented Gradients (HOG). Hence it is evident that to build a foolproof and robust feature-based tracking system, fusion of multiple cues in tracking framework is essential. In the proposed method, the color features which represent the target appearance and the edge oriented histogram features which describe the target structure are used to characterize the target uniquely in the scene. Further, the edge features are chosen for their ability to represent the target structure overcoming the errors occurring when there exists similar color models between the target and background and also for their significant robustness towards illumination variation. These features can be easily extracted and consume lesser computation time.

Some of the popular algorithms developed recently use learning-based approach for tracking. Kalal et al. proposed TLD [13], which decomposes the long-term tracking task into tracking, learning and detection. An alternate method, Staple [14], combines two image patches to learn a model, robust to color change and deformations. A framework for adaptive tracking using a kernelized structured output support vector machine, popularly known as STRUCK [15], was proposed by Hare et al. In order to reduce the learning-based computations, SRDCF [16] algorithm was proposed which tries to reduce the problems resulting from periodicity assumptions in learning correlation filters. ECO [17] is a recent tracker which tries to drastically reduce the parameters in the model and improve tracking robustness. As can be seen, more recently the trend has shifted towards accuracy oriented learning-based models for tracking which significantly increases the time complexity and computation costs. Further, some of the methods work only on high-end GPU thus failing to operate for real-time sequences on normal systems [18]. This has led to development of trackers such as SiamFC [19] and Kernelized Correlation Filter [20], which operate at frame-rates beyond real-time.

III. Particle Filter and Mean-Shift Trackers

In this section, the conventional particle filter and mean-shift tracking algorithms are discussed individually. Also, the scheme for integration of mean-shift in particle filtering framework is provided at the end of this section.

A. Tracking using Particle Filter Framework

Particle Filter [21] is often used when the posterior density and observation density are non-Gaussian. Sequential Monte Carlo estimation evaluates the posterior probability density function $p(s_k|Z_k)$ of state s_k, given the set of measurements Z_k up to time k [10]. In the Monte Carlo approach, samples are used to represent the probability density function (pdf) of the state. These multiple samples are known as particles which have an associated weight w_k^l, which signifies the quality of specific particle l, where $l = \{1, 2, ..N_s\}$ and N_s represents the total number of samples representing the posterior pdf. The problem of tracking a target can be formulated as,

$$s_{k+1} = f(s_k, \nu_{k+1}), \tag{1}$$

$$Z_{k+1} = h(s_{k+1}, n_{k+1}). \tag{2}$$

where f is the state transition model, h is the observation model and Z_k is the measurement at time instant k. ν and n represent the process and measurement noise, respectively. Estimate of a new state is computed using the weighted sum of particles. To generate this estimate, the two critical steps are $prediction$ and $update$.
Using the Bayesian framework, the conditional pdf $p(s_{k+1}|Z_k)$ is recursively modified as per $prediction$ step and $update$ step. The $prediction$ step is given as,

$$p(s_{k+1}|Z_k) = \int_{\Re^{n_x}} p(s_{k+1}|s_k)p(s_k|Z_k)dx_k, \tag{3}$$

and $update$ step is,

$$p(s_{k+1}|Z_{k+1}) = \frac{p(Z_{k+1}|s_{k+1})p(s_{k+1}|Z_k)}{p(Z_{k+1}|Z_k)}. \tag{4}$$

It is difficult to find a simple analytical expression for propagation of $p(s_{k+1}|Z_{k+1})$ through eq (4), hence we use numerical methods. In the particle filter framework, a set of weighted N_s particles are drawn from posterior conditional pdf. These particles are then used to map the integrals into discrete sums. The discretized posterior can be approximated as,

$$\hat{p}(s_{k+1}|Z_{k+1}) \approx \sum_{l=1}^{N_s} \hat{w}_{k+1}^l \delta(s_{k+1} - s_{k+1}^l), \tag{5}$$

where $\hat{w}_k^l$ is the normalized weight on each particle. It can be proved that as $N_s \to \infty$, the approximation in eq (5) approaches $p(s_{k+1}|Z_{k+1})$, which is the true posterior density [21].

B. Visual Tracking using Mean Shift

Mean shift [22] is a robust non-parametric approach used to find the modes of the probability distribution functions by climbing the density gradients. In a tracking scenario, if the target has its characteristics defined as density function q, then a position y in current frame can be located, which ensures that the characteristic candidate density function p_y is most similar to target characteristic density q. If the characteristics at location y have highest similarity, it implies that the target is located at position y. Usually in case of visual tracking, the most common measure to find similarity between two distributions is the Bhattacharya coefficient defined in eq (12). Expanding the Bhattacharya coefficient using the Taylor series approximation at initial position y_0 and using eq (10) give the following,

$$\rho[p_u(y), q_u] \approx \tfrac{1}{2}\sqrt{p_u(y_0), q_u} + \tfrac{C_h}{2}\sum_{i=1}^{I} w_i k_w \left(\left\|\tfrac{y-x_i}{h}\right\|^2\right), \tag{6}$$

where y_0 is the location of target in previous frame. In eq (6), the first term is independent of y, thus there is a need to minimize only the second term. Also in eq (6), the weight w_i is computed as,

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(y_0)}}\delta(h(x_i) - u) \tag{7}$$

At each iteration, the new target location y_1 from previous location y_0 is calculated as,

$$y_1 = \frac{\sum_{i=1}^{I} x_i w_i g\left(\left\|\tfrac{y-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{I} w_i g\left(\left\|\tfrac{y-x_i}{h}\right\|^2\right)} \tag{8}$$

where $g(.)$ is the derivative of kernel k chosen for the mean-shift procedure. If Epanechnikov kernel is employed [4], the derivative of this kernel is constant and hence eq (8) can be reduced to weighted distance average given as,

$$y_1 = \frac{\sum_{i=1}^{I} x_i w_i}{\sum_{i=1}^{I} w_i} \tag{9}$$

which corresponds to the new location of the target.

C. Integration of Mean Shift with Particle Filter Framework

Maggio et al. [3] proposed integration of mean-shift in particle filter framework which combined the individual advantages of the above two procedures. This approach termed MSEPF leads to more efficient sampling as it shifts samples to the neighboring modes, overcoming the degeneracy issue in particle filter. It also requires fewer number of particles to maintain multiple state hypotheses resulting in lower computational costs. The herding of samples to the nearest modes shifts the focus only on samples with larger weights. Since samples move to their neighboring local maxima actively after mean shift analysis, one does not require more samples to retain multiple modes. Even if there exists a higher number of samples, many will converge to the same mode due to mean shift iteration. Hence the retention of multiple modes in posterior density with fewer particles leads to enhanced computational efficiency as regards to conventional particle filter.

A visual representation of MSEPF algorithm is shown in Fig 1. Once the samples are propagated using the transition model $p(s_{k+1}|s_k)$ as shown in step no.2 of Fig 1, mean-shift optimization algorithm is applied on each sample. Particles are shifted in gradient ascent direction to their adjacent local modes until they converge. These shifted particles represent the modes of the distribution. Since local maxima are fairly represented, the multi-modal distribution is now represented with the aid of very few particles. Using this knowledge of maxima, a new particle set is generated using Importance Sampling [21] principle without destroying the original distribution as shown in step no.5 (i.e. Resampling) in Fig 1.

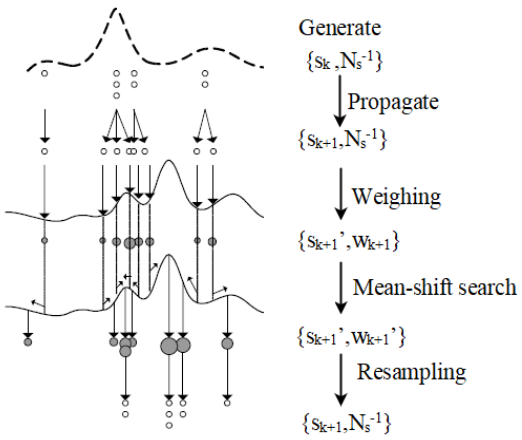


Fig. 1: Working of MSEPF algorithm.

Thus, it can be observed that the integration of mean-shift into particle filter allows tracking using fewer number of particles. The results using these fewer particles might be satisfactory for sequences with no challenges involved. However, for sequences with varied challenges such as occlusion, scale variation, fast motion etc., such tracker fails. The reason for this being insufficient number of particles available to hold the multiple and ever-changing state hypotheses in presence of challenges. This article precisely aims to overcome the above failure and tries to deal with challenge of occlusions with selected few particles. A strategy is devised which allows a smaller particle set to represent sufficient probable states when the target is occluded. Once occluded, a fast recovery strategy is also adopted. This additional ability incorporation makes the tracking more robust with lesser number of particles as compared to the conventional particle filter. Also, the use of both color and EOH features makes the tracking less prone to failures.

IV. Proposed method: Mean-Shift Occlusion Aware Particle Filter

In this section, the proposed Mean-Shift Occlusion Aware Particle Filter (MSOAPF) algorithm is presented. The method enables tracking of the target using a fewer number of particles, while addressing the challenge of occlusion faced during tracking. The MSOAPF uses fusion of color and EOH features to represent the target characteristics while tracking, as described in the rest of this section.

A. Features used for Tracking

In this section, the features which are used to uniquely represent the target in the scene are discussed. Two features, namely color model and EOH features for constructing the likelihood model are used while tracking. Color model has the ability to represent target appearance whereas the edge features can define the outline of the target, effectively. Hence, they are used together to uniquely characterize the target in the scene.

1) Color Distribution Model: As color models are robust to occlusion, rotation and non-rigidness [9] of the target, they are used as target models in tracking applications. To make the tracking more robust to changes in illumination, HSV space is chosen over RGB space [23]. Color histograms for region are generated using function $h(x_i)$ which will assign the color at location x_i to one among the discretized m-bins. Color distribution of the object at location y given as $p_y = \{p_u^{color}(y)\}_{u=1,2..m}$ is computed as,

$$p_u^{color}(y) = C_h \sum_{i=1}^{I} k_w \left(\left\| \frac{y - x_i}{h} \right\| \right) \delta[h(x_i) - u]. \quad (10)$$

where I is the number of pixels in the selected patch of image, δ is Kronecker delta function, $||.||$ is the norm, $k_w(.)$ is the weighing function allotting higher weights to pixels nearer to region center than others, $h = \sqrt{H_x^2 + H_y^2}$ and C_h is the normalization factor.

2) Edge Orientation Histogram: Whenever the color of the target is similar to background, a tracker solely based on the color model will track poorly. Thus, edge features are added to the tracker to represent the target contour. The EOH is constructed once all the target edge points are detected by the Sobel edge detector [24]. Edge detection is carried out by computing the horizontal and vertical derivatives obtained by convolving the grayscale image Im with 3x3 kernels. The pdf for edge for target candidate as discussed in [25] is given as,

$$p_u^{edge}(y) = C_e \sum_{i=1}^{j} k_w \left(\left\| \frac{y - x_i}{h} \right\| \right) \delta(b_e(x_i) - u), \quad (11)$$

where x_i is the center of the target candidate, j is the number of bins, $b_e(x_i)$ is the edge orientation bin in the quantized edge oriented space at location x_i and C_e is the normalization factor.

3) Similarity measure between features: To find similarities between these two densities p and q, a parameter known as Bhattacharya Coefficient is computed, defined for discrete densities as,

$$\rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}, \quad (12)$$

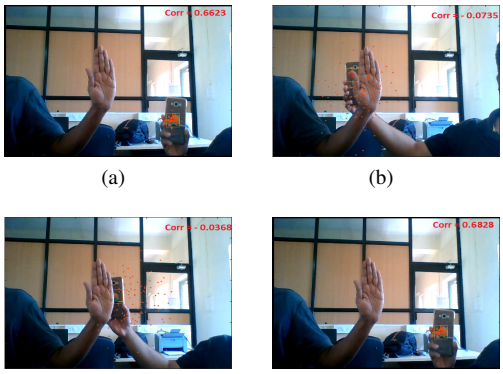
where $u = \{1, 2..m\}$ is the bin index of the discrete density. The discrete densities can be considered equivalent to histograms of color and edges considered in the proposed tracker. The color and EOH features are extracted from the target as well as probable candidates and similarity between them is computed using the Bhattacharya coefficient. Based on the Bhattacharya coefficient, the Bhattacharya distance can be defined as,

$$d[p(y), q] = \sqrt{1 - \rho[p(y), q]}. \quad (13)$$

Higher Bhattacharya coefficient refers to high degree of similarity between the two densities, with $\rho = 1$ indicating a perfect match between two densities.

B. Detection of Occlusion and Occlusion Recovery

Occlusion recovery strategy is to be adopted only after occlusion detection, otherwise the algorithm proceeds with normal tracking procedure. To detect occlusion, a 2D correlation function is used. The 2D correlation function $Corr$



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

Fig. 2: An illustration of particles following constant velocity model when target is not occluded. When target gets occluded, the particle search area is increased for target recovery.

between the selected target patch in the initial frame and image patch at filtered location using particle filter is computed as,

$$Corr = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}}. \quad (14)$$

where A and B are the two image patches of dimensions $m \times n$, and $\bar{A}$ and $\bar{B}$ represents the 2D mean of A and B, respectively. Higher values of correlation imply that the target is tracked properly, whereas lower values imply that the target is lost or occluded. An appropriate threshold τ for this correlation value is selected to detect occlusion. If $Corr < \tau$, then the target is occluded or lost.

Once occlusion is detected, the area of generation of particles around the center of the blob is increased radially outwards to engulf the entire area of the occluding object in order to identify the position from where the target might reappear in the scene. The motive of gradually increasing the search area by spreading out particles is to incorporate all the changes in direction and velocity that the target may undergo while being occluded. The goal is to recapture the target as quickly as possible once it re-enters the view. Hence, once the object gets occluded, the transition model is altered from constant velocity model to random walk model. This altered transition model increases the area for which the target is to be searched post occlusion. Once occluded, the switched transition model to update and distribute particles around x_k is given as,

$$s_{k+1} = s_k + \zeta_k. \quad (15)$$

where ζ_k is 2D uniform distribution in a circle with radius r, which is increased gradually outwards. An illustration of this occlusion detection and target recovery is shown in Fig 2. In Fig 2 (a), the target, which is the green ball, is clearly visible in the scene. The correlation value between the target and reference is 0.6623, which is greater than the chosen threshold value of 0.1. This implies that there is no occlusion and tracking proceeds with constant velocity motion model. The area of spread of particles is also very less as the ball is not occluded, which can be seen from Fig 2 (a). Now, in Fig 2(b), the target ball is fully occluded by the hand of a person. This can be deduced from the correlation value which is now -0.0735, which is less than the chosen threshold. In order to locate the position from where the ball will probably re-enter the scene, the motion model is switched and the area of particles is gradually spread outward using eq (15) so that it engulfs the entire hand until the target reappears in the scene. This can be seen in Fig 2(b) where the particles are spread in larger area to recapture the target post occlusion. Also, it is observed that the target ball changes its direction of motion when it is fully occluded by hand as can be seen from Fig 2(c). The target was initially moving leftwards in Fig 2(a) and (b) and once it is fully occluded, it changes its direction towards right and reappears from the opposite side of which it was expected to reappear. As soon as the target reappears and is detected (even partially), the correlation value increases, as can be seen in Fig 2(c), where the correlation value is -0.0368, which is still less than threshold. Hence, the spread of particles is still more. This continues till the target is fully located as shown in Fig 2(d), where the correlation value, now at 0.6828, is above the threshold value, leading to tracking with the constant velocity model with lesser particle spread.

The entire process can be summarized as: If $Corr < \tau$, that means occlusion is detected and the motion model given in eq (15) is used. Otherwise the constant velocity model is adopted for $Corr \geq \tau$.

The method also addresses the challenge of fast moving targets. The fast motions make the tracker lose track of the target resulting in lower values of correlation. The lower values of correlation during tracking implies that the target is not being tracked effectively and corrective measures ought to be taken. Using the same strategy of gradual area expansion of particles as discussed above, the fast moving targets, once missed can be recaptured and tracked effectively. Thus, the MSOAPF can handle both the problems of occlusion as well as fast moving target.

C. Working of MSOAPF Algorithm

We propose to integrate mean-shift in the particle filter framework and also to incorporate the fast failure recovery strategy to handle occlusions while tracking. A flowchart of the proposed method is illustrated in Fig 3.

The target to be tracked is selected by placing a bounding box around it following which the color and EOH features of the target within the selected blob are extracted. Initially, for the first iteration, the spatial centroid of this blob is considered to be the mean point around which N_s number of particles representing hypotheses are generated randomly within a specified width and height. After completion of first iteration, the spatial centroid is calculated using the weighted average of particles. Once generated, these samples are updated using the constant velocity motion model. Thus each of the N_S particles representing state hypotheses is defined as,

$$\{s_k^l\}_{l=1}^{N_s} = \{x_k^l, y_k^l, \dot{x}_k^l, \dot{y}_k^l\}. \quad (16)$$

where (x_k, y_k) denotes the centroid of the blob, while $\dot{x}_k$ and $\dot{y}_k$ are x and y directional velocities of the target, respectively.

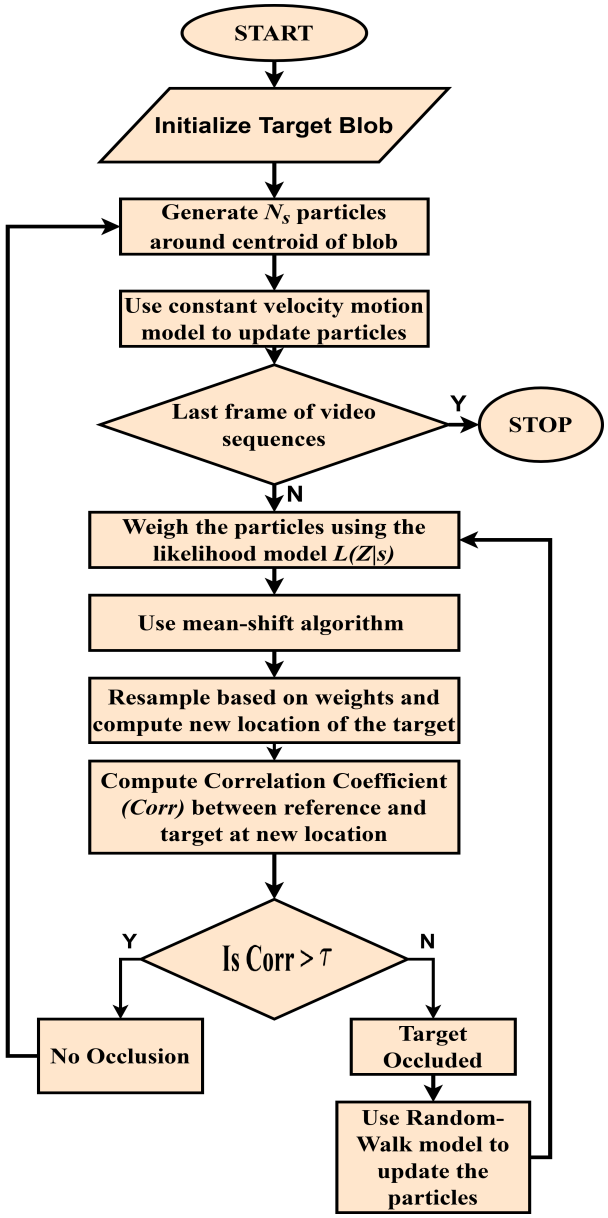


Fig. 3: Flowchart illustrating the working of proposed algorithm.

The dynamic model used for transition of particles from one state to another is represented as,

$$s_{k+1} = Fs_k + \mu_k. \tag{17}$$

where F is the transition matrix of the motion model and μ_k is a multivariate Gaussian random variable.

After updating the particles using the transition matrix, mean-shift is applied to each particle. Mean-shift algorithm moves them in gradient ascent direction as per their observation likelihoods, until each of them converges to their adjacent local maxima. This set of particles now represent modes of the multi-modal distribution. The particle set now corresponds to probable candidate positions where the target may have moved after the update step. Considering these locations as the updated centroid locations of the target, the color and EOH features are extracted at the updated particle locations. For each of the N_s particles, color and EOH features are extracted. These features from probable candidates are then compared for similarity with the features of the target extracted in the initial frames. The distance between two feature distributions is computed using the Bhattacharya distance which determines the quantum of weight to be assigned to each particle. The weighing of particles is based on the observation likelihood model which is characterized by the Gaussian distribution as,

$$L(Z|s) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{d^2}{2\sigma^2}\right), \tag{18}$$

where d corresponds to Bhattacharya distance computed in eq (13). We consider the same distribution of weights as shown in eq (18) for both color and EOH features. Bhattacharya distance for color and EOH features is computed individually and assuming that likelihood of color and edge are statistically independent of each other, the multi-cue likelihood of both features is fused as,

$$L(Z|s) = L_{color}(Z_{color}|s)L_{edge}(Z_{edge}|s). \tag{19}$$

Based on the likelihood model given in eq (19), weight w is assigned to each particle as,

$$\{w_{k+1}^l\}_{l=1}^{N_s} \propto L(Z|s), \tag{20}$$

These weights are then normalized, using which, the new state hypothesis is computed by the weighted average method as,

$$s_{k+1} = \sum_{l=1}^{N_s} w_{k+1}^l s_{k+1}^l. \tag{21}$$

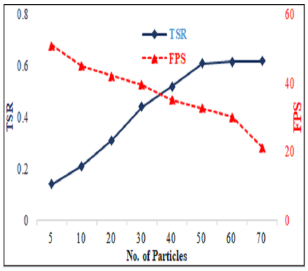
At this new position obtained by eq (21), N_s particles are again generated. Also, 2D correlation is computed between the reference blob and the blob at new position obtained in eq (21) to detect occlusion. The value of 2D-correlation obtained is compared with the threshold as to decide whether occlusion has occurred or not. If the object is occluded, the motion model is modified as shown in eq (15), else the algorithm proceeds with the constant velocity model.

V. Experimental Setup and Dataset Used

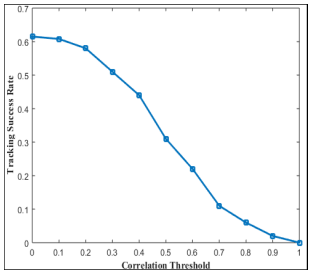
For analysis of the results, the proposed algorithm is executed on three benchmark datasets: Visual Tracking Benchmark ($OTB100$) [26], Visual Object Tracking 2018 (VOT18) [27] [28] and TrackingNet [29]. All the algorithms are executed on an Intel Core i7-7700 CPU system with 16 GB RAM and 3.60 GHz Personal Computer. The algorithm is implemented in MATLAB $R2016a$. The number of particles N_s for the proposed scheme is chosen as 50, based on the obtained result for Tracking Success Rate as illustrated in Fig 4 (a). Similarly, based on the Tracking Success Rate, the experimental threshold for 2D correlation τ is fixed to be 0.1 as shown in Fig 4 (b).

VI. Results and Analysis

The experiment begins with determining the number of particles needed for tracking. For this purpose, the performance of the tracker is observed for sequences from $OTB100$ dataset. The number of particles is varied for each observation and the minimum number of particles which gives higher tracking success rate is considered. The plot comparing the Tracking



(a) TSR and FPS for different number of particles



(b) Plot of TSR vs correlation threshold.

Fig. 4: Selection of number of particles and correlation threshold

Success Rate (TSR) [1] and the execution speed in terms of Frames Per Second (FPS) for different number of particles considered for MSOAPF is shown in Fig 4 (a). It can be observed that as the number of particles approach 50, the TSR value stabilizes itself and remains constant even when the number of particles is increased for MSOAPF. Hence, the number of particles for the purpose of this experiment is chosen to be 50. Also, for 50 particles, the FPS is 38.2, making the algorithm suitable for real-time applications. Next step is to determine the value of correlation threshold to detect the occurrence of occlusion. For this purpose, the plot of correlation vs TSR is shown in Fig 4 (b). The correlation value is considered in steps of 0.1 in the range 0 to 1. For each value, the TSR is plotted in Fig 4 (b). It can be observed that for values below 0.1 the TSR is almost constant. Thus, inference is drawn to fix the correlation value to be 0.1.

The proposed method is compared with fourteen state-of-the-art tracking algorithms: DSST [30], TLD [13], ECO [17], STRUCK [15], KCF [20], IVT [31], STAPLE [14], SiamFC [19], MDNet [32], SRDCF [16], SAMF [33], SiamRPN [34], DNT [35] and TADT [36]. Despite being a iterative feature-based object tracking approach, we have compared MSOAPF with advanced state-of-the-art learning-based approaches such as ECO [17] and MDNet [32] in order to illustrate the performance.

A. *Evaluation Methodology*

To evaluate the performance, precision plots and overlap success plots are provided. Precision plot indicates the percentage of frames for which the predicted location is within the chosen threshold distance, 20 pixels in this case, from ground-truth. These plots are indicated for the entire dataset as well as for various challenge-specific sequences like scale variation, occlusion, background clutter, illumination variation and fast motion which are encountered while tracking. The success plot shows ratio of successful frames at different thresholds varied from 0 to 1. The AR-raw plot [28] generated is also provided for ranking of different algorithms executed on $VOT18$ dataset. Also, the performance based on the processing speed for an algorithm is measured in units FPS.

B. *Result Analysis of $OTB100$ Dataset*

The analysis begins by comparing the performance of proposed MSOAPF (color + EOH features) with MHOG

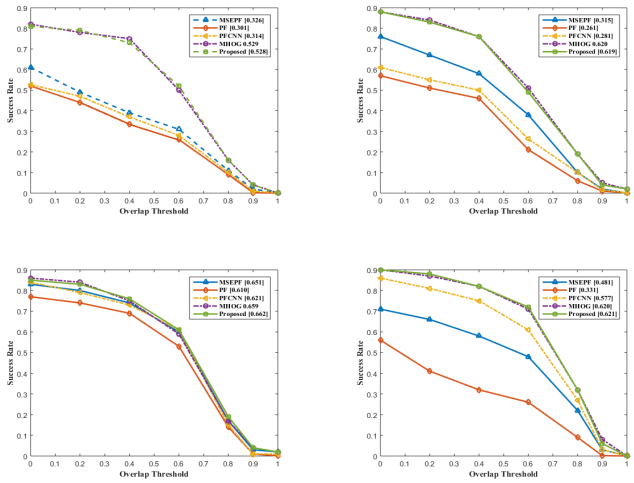


Fig. 5: Comparison of Success plot for different challenges: Occlusion, Fast motion, Background Clutter and Illumination Variation (clockwise).

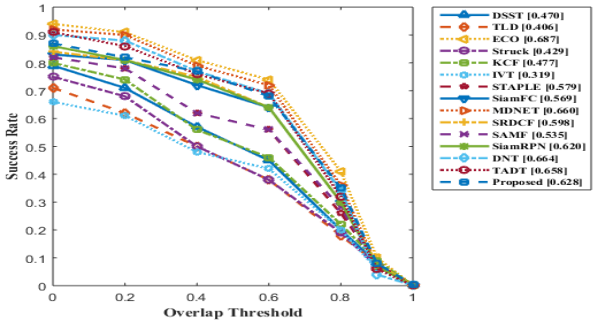


Fig. 6: Overlap Success Plots for all the sequences from $OTB100$ dataset

(MSOAPF with color and HOG features - 50 particles), MSEPF (50 particles), conventional particle filter (200 particles and HOG feature) and particle filter using CNN features (PFCNN) (50 particles) for sequences with different challenges from $OTB100$ dataset. The challenges considered for comparison are of occlusion, fast motion, illumination variation and background clutter. The success plots for different challenges are shown in Fig 5. From these figures, it can be observed that for illumination variation, the performance of MSEPF and MSOAPF is almost comparable. This is due to the simple motion of object which does not undergo hefty changes in its motion. However, the MSOAPF outperforms all the trackers in case of occlusion and fast motion. The simple motion update models which cannot anticipate fast and abrupt motions leads to tracking failures in case of conventional particle filter and PFCNN. Also, the inadequate target representation leads to failures in case of MSEPF. However, the MSOAPF robustly tracks the target and gives better results than the above three algorithms. It can also be seen from these plots that MSOAPF can track efficiently using just 50 particles as compared to conventional particle filter with 200 particles. It can also be observed that the proposed MSOAPF (with EOH features) and MHOG give similar and comparable performance without much variation in the outcome of the experiments.

Next, the success plot and precision plot for the entire $OTB100$ dataset is shown in Fig 6 and Fig 7, respectively.

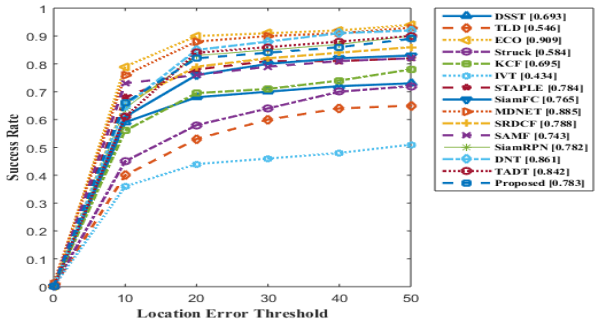


Fig. 7: Precision plots for all the sequences from $OTB100$ dataset

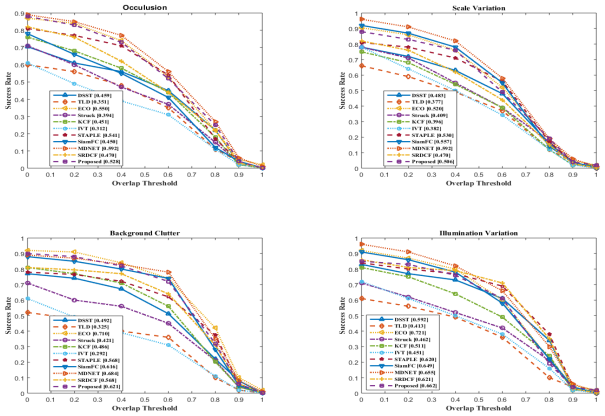


Fig. 8: Success plot for different challenges: Occlusion, Scale Variation, Illumination Variation and Background Clutter (clockwise)

As can be seen from the plots, the proposed MSOAPF ranks fifth in case of comparison with other considered methods. The methods which rank higher are ECO, MDNet, TADT and DNT. This is due to the fact that all these four methods use learning-based tracking approach whereas the MSOAPF uses iterative feature-based tracking approach. However, for such learning based approaches, the execution time in terms of FPS is much high as can be seen from Table I. MSOAPF tracker performs faster due to absence of neural networks or CNNs Also, MSOAPF does not involve training of system with target information and characteristics. The absence of all such steps reduces the computational cost greatly without compromising on the tracking performance. For every one unit of processing time for MSOAPF, 0.162 unit is used for feature extraction, 0.472 goes for feature matching, 0.254 is for mean-shift whereas 0.112 is used for particle filtering. The KCF tracker, although yielding higher FPS than MSOAPF, tracks the target poorly as can be seen from precision and overlap success plots. The MSOAPF algorithm despite ranked fifth overall, performs better than other trackers which involve learning. At some instances the MSOAPF does fail but due to fast recovery strategy, the tracker relocates the target much faster than other trackers.

In Fig 8, the challenge-wise success plot for challenges of background clutter, occlusion, scale variation and illumination variation for sequences from $OTB100$ dataset is shown. The MSOAPF gives comparable performance to other methods and is always within the top five ranks for challenge of scale variation, background clutter, illumination variation and

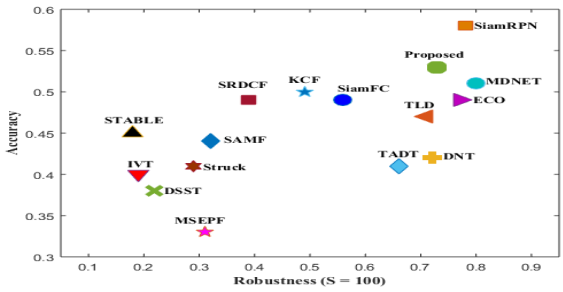


Fig. 9: AR- raw plot and comparison for $VOT18$ dataset

occlusion. Thus, the tracker gives comparable performance with other state-of-the-art trackers for these challenges. This is due to the considered feature set for target description and quick occlusion detection and recovery strategy, which enables faster relocation of the target.

C. Result Analysis of VOT2018 Dataset

In this section, the results obtained for the $VOT2018$ dataset are presented, which has 60 challenging video sequences. The performance is evaluated based on the robustness and accuracy of the tracker. As can be observed from AR-raw plot in Fig 9, the tracker is ranked fourth for robustness. The MSOAPF tracker has a good balance of both accuracy as well as robustness. It is re-iterated that trackers ranked above MSOAPF are learning based trackers which enables them to achieve higher ranks in robustness and accuracy. These trackers have a low tracking speed and some even require GPU for performing in real-time applications [18]. However, the proposed tracker is implemented on CPU and is found to be suitable for real-time applications. Further, for $VOT18$ dataset, comparison of different algorithms in terms of tracking failures per 100 frames for challenge-specific sequences is shown in Fig 10. An interesting thing to note is that MSOAPF ranks first for the challenge of occlusion which is closely followed by ECO and MDNet. Similar performance is observed for motion change and camera motion challenges, where the target either changes its motion abruptly or appears to have changed its motion due to movement of camera. Even for these challenges, MSOAPF algorithm ranks among the top three trackers due to the quick recovery achieved by switching of motion model when the tracker detects the loss of track.

D. Result Analysis of $TrackingNet$ Dataset

The success plot and precision plot for sequences from $TrackingNet$ dataset are presented in Fig 11 and Fig 12, respectively. As was observed in case of $OTB100$ dataset, similar results are obtained for $TrackingNet$ dataset. It can be observed that the tracker overall ranks fourth and gives comparable performance with other state-of-the-art trackers.

E. Visual Analysis of Results

This section presents visual results of the proposed scheme on different sequences. Due to space issues, only two sequences, one each from $OTB100$ and $VOT18$ dataset are shown which have heavy occlusion present in them. In Fig

TABLE I: Comparison of Mean FPS for different methods

	TADT	TLD	MSEPF	STRUCK	KCF	DSST	STAPLE	SRDCF	ECO	SOD-LT	SiamRPN	MDNET	DNT	IVT	SiamFC	Proposed
Mean FPS	33.7	23.6	39.1	10.4	110.4	11.9	21.8	3.12	4.16	5.21	141.6	0.714	3.2	11.7	58.0	38.2

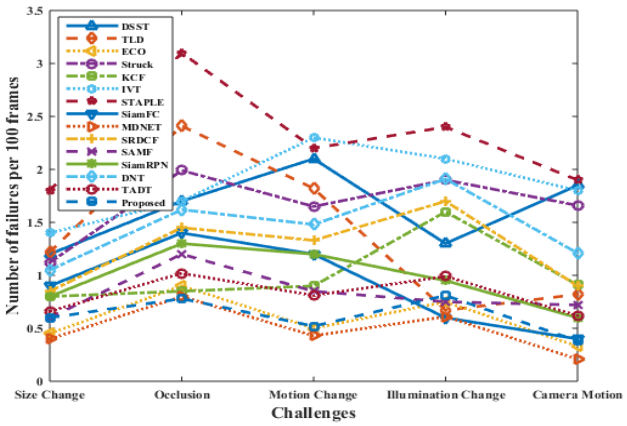


Fig. 10: Plot showing number of failures per 100 frames for different challenges involved in $VOT18$ dataset.

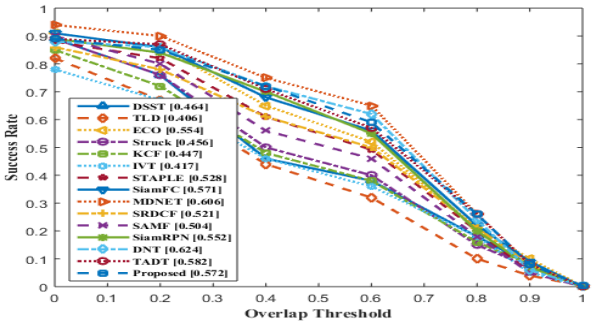


Fig. 11: Overlap Success Plots for sequences from $TrackingNet$ dataset

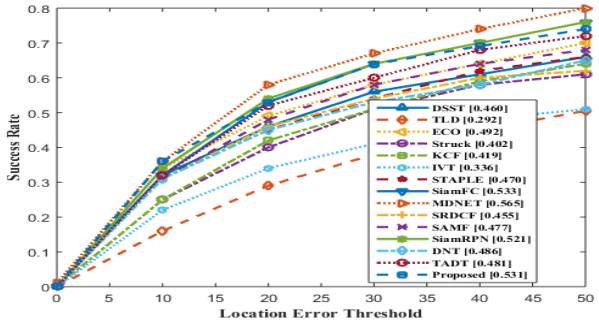
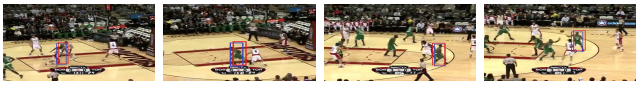


Fig. 12: Precision plots for all the sequences from $TrackingNet$ dataset



(a) Tracking results for *basketball* sequence from $OTB100$ dataset
Frame No. 17, 64, 489, 620

(b)Tracking results for *girl* sequence from $VOT18$ dataset
Frame No. 106, 131, 221, 305

Fig. 13: Comparison of results for *basketball* and *girl* sequences.

13 (a), the goal is to track the basketball player in green jersey who is at certain times occluded by his teammates and opponents. In frame 17, he is occluded by opponent despite which he is tracked successfully in consecutive frames. Between frames 300 and 489, he is occluded by his own player wearing jersey of same color. However, due to robust feature set and due to motion model incorporated, the player is continuously tracked as seen in frames 489 and 620. In Fig 13 (b), the target is a girl who is occluded by an man walking with a bicycle. The color of the girl's dress matches with the color of the cycle despite which the girl is tracked without any failure. Between frames 106 and 131, the girl is fully occluded by the man. During occlusion, the tracker tries to relocate the target, which it successfully does after the girl reappears in the scene as can be observed in frames 131, 221 and 205.

VII. Conclusions

A mean-shift occlusion aware particle filter (MSOAPF) is proposed in this article to track a target with fewer number of particles while being able to address the challenge of occlusion. The mean-shift operation allows the particle to be set to represent the modes of the distribution, enabling tracking with fewer number of particles as compared to a conventional particle filter. If there is long term occlusion, the situation is first detected through means of correlation and then an occlusion recovery strategy is adopted wherein the motion model of the particles is switched to random walk model for faster recapturing of the target. The use of simple color and EOH features for target description yields higher execution speed as compared to the other algorithms which use complex feature set, which also leads to lag in video sequences when executed on CPU.

Performance evaluation is carried out on three benchmark datasets $OTB100$, $VOT18$ and $TrackingNet$ and the results are compared with fourteen existing state-of-the-art tracking algorithms including advanced-learning based trackers. The MSOAPF has ability to process approximately 38 frames per second on CPU thus making it suitable for real-time tracking applications. For occlusion-specific cases, which is the focus of this article, the tracker outperforms even the learning-based trackers. For other challenges such as illumination variation, background clutter and fast motion, the tracker gives comparable accuracy to advanced trackers and in certain cases is ranked above learning-based approaches. Considering the fact that MSOAPF is an iterative feature-based tracking approach, the results obtained are comparatively better.

The proposed MSOAPF algorithm has the ability to handle occlusion and quickly recapture the target once it reappears post-occlusion. The advantages of the proposed method are its easy implementation on a simple CPU and its applicability for real-time videos. Further, tracking is carried out using only 50 particles as against conventional particle filter which requires around 200 particles for holding state hypotheses. Also, tracking is carried out using the target features sans any training step, as is done in learning-based trackers.

The MSOAPF focuses mainly on the challenge of occlusion. However, there are other challenges such as illumination variation, background clutter, etc. which are required to be addressed. The ability of the MSOAPF tracker to handle these challenges needs to be enhanced. Also, a mechanism to dynamically fix the correlation threshold parameter needs to be explored.

REFERENCES

[1] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[3] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2005, pp. 221–224.

[4] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, 2007.

[5] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013.

[6] K. Meshgi, S. Maeda, S. Oba, H. Skibbe, Y.-z. Li, and S. Ishii, "An occlusion-aware particle filter tracker to handle complex and persistent occlusions," *Computer Vision and Image Understanding*, vol. 150, pp. 81 – 94, 2016.

[7] Z. Duan, Z. Cai, and J. Yu, "Occlusion detection and recovery in video object tracking based on adaptive particle filters," in *Proceedings of IEEE Chinese Control and Decision Conference*, 2009, pp. 466–469.

[8] S. D. Lin, J.-J. Lin, and C.-Y. Chuang, "Particle filter with occlusion handling for visual tracking," *IET Image Processing*, vol. 9, no. 11, pp. 959–968, 2015.

[9] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.

[10] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential monte carlo tracking by fusing multiple cues in video sequences," *Image and Vision Computing*, vol. 25, no. 8, pp. 1217–1227, 2007.

[11] M. Niu, X. Mao, J. Liang, and B. Niu, "Object tracking based on extended SURF and particle filter," in *Proceedings of International Conference on Intelligent Computing*, 2013, pp. 649–657.

[12] S. Rahimi, A. Aghagolzadeh, and H. Seyedarabi, "Human detection and tracking using new features combination in particle filter framework," in *Proceedings of 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, 2013, pp. 349–354.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[14] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "STAPLE: Complementary learners for real-time tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409.

[15] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "STRUCK: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[16] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[17] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: efficient convolution operators for tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6638–6646.

[18] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 76, pp. 323–338, 2018.

[19] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 850–865.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[21] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[22] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *Proceedings of International Conference on Image Processing*, vol. 3, 2000, pp. 70–73.

[23] L. Shuhua and G. Gaizhi, "The application of improved HSV color space model in image processing," in *Proceedings of 2nd International Conference on Future Computer and Communication*, vol. 2, 2010, pp. V2–10–V2–13.

[24] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[25] Y. Chia, W. Y. Kow, W. L. Khong, A. Kiring, and K. T. K. Teo, "Kernel-based object tracking via particle filter and mean shift algorithm," in *Proceedings of 11th IEEE International Conference on Hybrid Intelligent Systems (HIS)*, 2011, pp. 522–527.

[26] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418.

[27] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.

[28] M. Kristan, A. Leonardis, J. Matas, and et al., "The sixth visual object tracking vot2018 challenge results," in *In Proceedings of Europenan Conference on Computer Vision 2018 Workshops*. Cham: Springer International Publishing, 2019, pp. 3–53.

[29] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 300–317.

[30] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of British Machine Vision Conference*, 2014.

[31] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[32] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[33] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of European Conference on Computer Vision*. Springer, 2014, pp. 254–265.

[34] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[35] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking." *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

[36] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1369–1378.