IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 June 2024, accepted 6 July 2024, date of publication 15 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3428321

RESEARCH ARTICLE

Particle Filter-Based Video Object Tracking Scheme With Target Remodeling and Reinitialization and Its Hardware Implementation Using Raspberry Pi

JYOTIRANJAN PANDA[1], (Member, IEEE), PRADIPTA KUMAR NANDA[1], (Senior Member, IEEE), AND TUHINANSU PRADHAN[2]

[1]Image and Video Analysis Laboratory, Department of ECE, Siksha 'O' Anusandhan, Deemed to be University, Bhubaneswar, Odisha 751030, India
[2]Department of CIOT, Siksha 'O' Anusandhan, Deemed to be University, Bhubaneswar, Odisha 751030, India

Corresponding author: Jyotiranjan Panda (jyotiranjan.iter@gmail.com)

This work was supported by Siksha 'O' Anusandhan, Deemed to be University.

ABSTRACT Video object tracking in real-world scenarios is one of the challenging problems of computer vision. The issue is compounded in the presence of varying illumination conditions, dynamic entities of the background, and bad weather conditions. In this research, a particle filter based new video object tracking scheme is developed with the proposed notions of target remodeling and reinitialization. During the tracking phase, the target is remodeled in each frame to take care of the changing scene dynamics over frames. The target is remodeled by fused feature distributions chosen from the created bank of fused feature distributions having discriminating potential to differentiate the target and the background in a given frame. The fused feature bank is created by fusing two features from the set consisting of Color, LBP, and HOG features. The features are fused probabilistically where the weights are determined based on the discriminating ability of a given feature. In order to achieve high tracking accuracy, the deviation of the tracker is evaluated in each frame using the notion of time motion history while reinitialization of the tracker position takes place when the deviation is above a preselected threshold. Besides, the proposed algorithm has been implemented successfully on a Raspberry Pi based hardware setup and thus becomes a potential candidate for real time implementation. The proposed scheme is successfully tested on videos from DAVIS 2016, LASIESTA, OTB 100, and CDnet 2014 databases and in most of the cases the tracking accuracy is found to be higher than those of the existing algorithms.

INDEX TERMS Particle filter, feature fusion, feature switching, target remodeling, target mean state reinitialization.

I. INTRODUCTION

Tracking of moving objects in a video sequence is one of the challenging problems in machine vision applications. The challenges are compounded due to the nonlinear motion of the object along with various scene complexities which arise because of illumination variation, clutter background, camouflage, occlusion, and dynamic background.

In a probabilistic framework, an object in each frame of a video sequence is modeled by its state distributions. Hence, the object motion in a video sequence is equivalent to the spread and the diffusion of these state distributions over time

The associate editor coordinating the review of this manuscript and approving it for publication was Long Xu.

frames [1], [2]. So, tracking the moving object trajectory is tantamount to finding the solution to the evolution of these state distributions over the frame sequence [1], [2], [3], [4], [5]. Since these state distributions are non-Gaussian in nature, the spread and diffusion occurring because of the object's motion are also nonlinear. With such challenges in a complex scene, the particle filter, which relies on the Monte Carlo chain framework and Bayesian probability, provides an excellent solution to this motion model of state distributions in discrete state space [5], [6].

Based on different issues of the scene complexity, different methods of tracking such as feature-based tracking [7], [8], [9], [10], model-based tracking [11], [12], [13], [14], [15], [16], [17], region-based tracking [18], and deformable template-based tracking [19], [20], [21] have been proposed. It is found from the literature that illumination variation and occlusion are the two crucial issues of the scene during object tracking [7], [8], [9], [10], [22], [23], [24], [25]. In case of occlusion, incomplete feature extraction of the target object at the occluded region leads to unstable tracking, whereas a sudden change of intensity in the scene due to illumination variation results in distortions in feature distribution thus resulting in an improper target model. An efficient attention networks (EAN) based visual tracking algorithim is proposed by Gu et al. [23]. The EAN module comprises a fused feature algorithim module (FFAM) where multi-level features are fused to enhance the discriminative attribute of the target model. Further, Gu et al. [22] have extended their work to handle target scale changes, occlusion and fast movement where a fused feature based new model is used to integrate the features of parallel branches to improve tracking accuracy. A multi-level feature enhancement unit together with a global channel attention network is proposed by Gu et al. [25] to strengthen the target model. Yuan et al. [24] proposed a video object tracking scheme for thermal infrared target tracking. Hence, researchers have been motivated to handle these two key issues of the scene by adapting the notion of fusion of multiple potential features to model the target in a particle filter framework. By and large, in different particle filter based scheme, the target model of initial frame is used over subsequent frames for object tracking. But it is to be noted that, the scene conditions change in the video due to various factors and hence use of the same target model althrough the tracking process may not be the appropriate model over the entire video. To take care of change in scene dynamics, we remodel the target object at each frame with appropriate fused features selected from the feature bank.

The concept of feature fusion in the particle filter framework has been adopted by some researchers to address the specific issues of the scene [7], [8], [9]. The issue of 3D-Human motion tracking is carried out by modeling the target with a fusion of contour and edge features [26]. Tracking in a cluttered environment is achieved by the fusion of shape and texture features [27]. In case of multiple object tracking, Younsi et al. [28] have fused multiple features such as shape, intensity, texture, and motion to model the target for effective tracking. Despite achieving good tracking capability, there are challenges to develop schemes for tracking the object under different complex dynamics of the scene. We have specifically considered scenes with illumination variation and dynamic entities in the background. These dynamic entities are due to bad weather conditions, background with snowfall, dynamic shadows, and splashing of water etc. The above mentioned research works focussed on enhancing the target model, but the initial target model is used in subsequent frames for tracking. In our approach, we have remodeled the target in subsequent frames with the selected fused feature distributions from the created fused feature bank to take care of the changing scene dynamics. Further, while remodeling, it is constrained that the remodel of the target does not deviate substantially from the original target model. Due to the above mentioned situations, even after efficient object modeling, the tracker may deviate beyond the acceptable limit of the mean position. This error could be cumulative in subsequent frames resulting in complete failure. In order to improve the tracking accuracy in such cases, reinitialization of the mean position of the tracker is proposed based on the notion of time motion history of the target.

In this paper, we propose a novel tracking scheme to track the video object in complex environments arising due to the above mentioned conditions. By and large, the target is modeled initially with appropriate features and the same model is used in subsequent frames for tracking. But due to the changing scene dynamics over the scene, the same model may not be appropriate for subsequent frames and may result in tracking errors. This tracking error of a given frame propagates and increases further resulting in complete failure of tracking the object. This motivated us to develop a particle filter based tracking scheme where remodeling of the target is carried out in every frame by fusion of appropriate features to achieve effective modeling of the target in the given frame. This is achieved by creating a fused feature bank and choosing an appropriate fused feature distribution by the proposed notion of switching of features. The fused features having discriminating attributes to differentiate the object and background in a given frame are chosen from the feature bank. Besides, while remodeling, it is taken care that the remodeling of the target at a given frame should not deviate much from the original target model. This is achieved by proposing an adaptive thresholding strategy for comparing the current model with that of the original target model. This notion improved the tracking accuracy but failed in some complex environments. To improve further, a new notion of a reinitialization of the mean tracker position is proposed to prohibit the tracker to deviate substantially from the object. This reinitialization of the mean tracker position is achieved using the joint notion of the time motion history of the frames and the discriminating feature of the fused feature distribution. The combined notion of remodeling of the target and reinitialization is found to track the target object under

different complex scenarios. The salient contributions of the proposed work is summarised as follows.

- At a given frame, fused feature distribution having discriminating attribute is used to model the target. For efficient target modeling at each frame, a bank of fused feature distributions is created and the most befitting one is selected for modeling.
- An adaptive threshold strategy is proposed to maintain the proximity of the target model of a given frame with that of the original target model.
- A novel strategy is proposed to reinitialize the mean state of the particle filter or in other words the mean tracker position using jointly the time motion history of the object and the discriminative attribute of the fused feature distribution.
- The proposed scheme is successfully implemented in Raspberry Pi based hardware systems.

Different benchmarked data sets such as CDnet 2014 [29], OTB 100 [30], LASIESTA [31] and, DAVIS 2016 [32] data sets are used to test the efficacy of the proposed algorithm. In all the above cases, the proposed algorithm tracked the objects with tracking accuracy which is either comparable or better than those of the existing ones.

The rest of the paper is organized as follows. Related works are presented in Sec. II. Section III presents the proposed tracking framework.The details about the target modeling using discriminating feature fusion is presented in Sec. IV. Reinitialization of the target location is presented in Sec. V and the remodeling of the target in each frame is dealt in Sec. VI. Tracking using particle filter is presented in Sec. VII. Details about Hardware implementation is presented in Sec. VIII whereas, the experimental conditions and data sets are presented in Sec. IX. Sec. X deals with the different results obtained and the corresponding discussions. The conclusions along with the future scope of work are presented in Sec. XI.

II. RELATED WORK

In particle filter based video object tracking schemes, target object modeling plays a crucial role for accurate tracking of the object in different frames of video. Different particle filter based video trackers have used different features to achieve effective tracking. It is found from the literature that single feature based object models such as color feature [33], [34], [35], edge feature [36], motion feature [37], [38], [39], appearance feature [40], [41] and entropy feature [42], [43] could handle one or two typical issues of the scene during tracking and have achieved good tracking accuracy for initial frames of the video. But, these trackers with single feature based object modeling fail to track the object after some frames due to the occurance of scene complexities in subsequent frames.

The issues of occlusion and illumination variation are handled simultaneously by the notion of feature fusion in modeling the target object. The problem of occlusion is taken care by Bhatt et al. [44], [45] where they have used the fusion of color and Edge-Oriented-Histogram (EOH) feature for accurate tracking. Additionally, the appearance and motion information of the object are used [46] to adaptively estimate the number of particles in each frame for accurate object tracking. In this scheme, a normalized correlation filter is integrated to the sparsity based tracker to improve the performance of tracking. The problem of people counting is addressed by García et al. [47], where the motion and height information are used to model the object in their proposed extended condensation algorithim. This algorithim is designed to solve the problem of occlusion without counting other objects like shopping trolleys. Talha and Stolkin [48] have proposed a particle filter based tracking scheme specifically for camouflaged targets by fusing the thermal and visible spectra camera data. The weights in the feature fusion process are determined based on the discriminating attribute in each frame. Besides, color and texture are also fused to model the target [35] in the Genetic algorithim based particle filter where Genetic algorithim is used for resampling. This helps in reducing the number of particles of particle filter while achieving improved tracking accuracy. In order to improve the tracking accuracy under occlusion and sudden illumination changes, Wang et al. [42] have fused the color and local entropy feature for target modeling. Specifically, the issue of occlusion is taken care by anti-occlusion particle filter proposed by Huan et al. [8] where the color and LBP features are fused with respective coefficients that are determined based on the difference between the object features and background.

The problem of object tracking under illumination variation of the scene is also taken care by Lu et al. [49] by their proposed object contour tracking based particle filter scheme. In this work, object rough location is realized by fusion of color histogram and Harries corner features. This scheme is very effective in object location and contour tracking. Further, Xu and Zhao [50] have proposed a particle filter based tracking algorithim which is based on the notion of adaptive fusion of color histogram and EOH. Using the notion of integral image frame, the authors have attempted to reduce the computational burden and enhance the tracking accuracy under illumination variation and partial occlusion condition. Besides, color and texture features are also fused by Ding et al. [51] and Panda and Nanda [6] for target modeling in the particle filter framework to enhance the tracking accuracy under illumination variation. It is found from literature that most of the particle filter based tracking schemes use a single target model throughout the tracking process. Hence, in this research target object is remodeled in each frame to take care of varying scene dynamics.

III. PROPOSED TRACKING FRAMEWORK

The block diagrammatic representation of the proposed scheme is shown in Fig. 1. As observed from Fig. 1, it has two important phases; (i) Initialization phase, and (ii) Target remodeling and tracking phase. The target remodeling and

tracking phase again consists of the diffusion of particles, the remodeling and tracking phase. Further, as observed from Fig. 1, the target remodeling phase includes the reinitialization of the mean state of the particle or in other words reinitializing the tracked location of the object in a given frame. After initializing the target model and the particles, the target is tracked in subsequent frames. Remodeling of the target is carried out in each of the subsequent frames and reinitialization of the tracked location takes place as and when necessary. Thus, tracking of the object is achieved in all the frames.

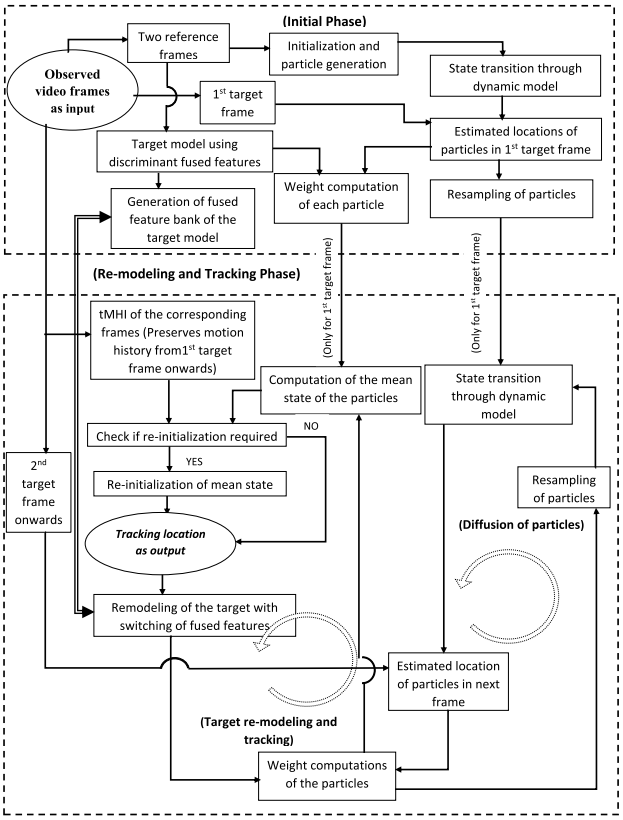

FIGURE 1. Block diagrammatic representation of the proposed scheme for object tracking.

A. INITIALIZATION PHASE

In beginning, two initial frames which are considered to be two reference frames are used to generate a particle which is replicated to generate N number of particles. These particles are considered in the particle filter based tracking scheme and correspond to possible tracker locations. In order to track the object in the third frame i.e. the first target frame, the particles pass through the dynamic model as shown in Fig. 2 to obtain the estimated locations in the target frame. Each particle in the target frame is weighted and the weight is computed by comparing it with the target model which is modeled by fused discriminant features. It is to be noted that the original target object is also modeled with different sets of distributions of discriminant fused features. All these

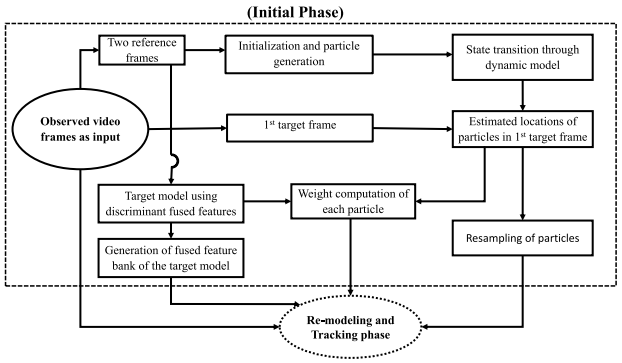

FIGURE 2. Block diagrammatic representation of Initialization phase of the proposed scheme.

fused feature distributions based models are stored in the fused feature bank. In our case, we have considered three discriminant features and hence developed three sets of fused feature distributions as the three different models of the target.

B. PROPOSED RE-MODELING AND TRACKING PHASE

Since the scene is expected to change from frame to frame, the scene dynamics need to be taken care of in each frame for effective modeling of the target.

1) REMODELING OF THE TARGET

Initially, for the target object, discriminating features are selected based on the target area of the template and the background area of the extended template. With the available discriminating features, two features are fused to generate fused feature distribution. In the process, a set of fused feature distributions is generated to create the fused feature bank as shown in Fig. 2. It may be noted that, different sets of fused feature distributions may be required for modeling the target in each frame. This is achieved based on the proposed notion of switching of fused feature distributions as shown in Fig. 3.

2) REINITIALIZATION PHASE

Besides depending on the scene dynamics, it may so happen that the tracker may deviate from the target to a great extent and thereafter may completely fail to track the object. To bring back the tracker to the actual position, the mean position of the tracker may be reinitialized. This is achieved by integrating the timed motion history ($tMHI$) of each frame with that the discriminating attributes of the features. It is assumed that the target movement is predominant in the scene. If the number of motion pixels in the tracker is less than a threshold, then it is considered as the deviation of the tracker from the object and hence the tracker position needs to be reinitialized. Since the tracker position is the mean state of the particles, the mean state of the particles is reinitialized as and when necessary. This is also reflected in the block diagram of Fig. 3. As seen from Fig. 3, for the first target frame, the mean state is computed and checked together with the $tMHI$ [52], [53] whether reinitialization is required or not during

the remodeling phase. The tracker location is determined and the target is remodeled with the notion of switching of feature distribution.

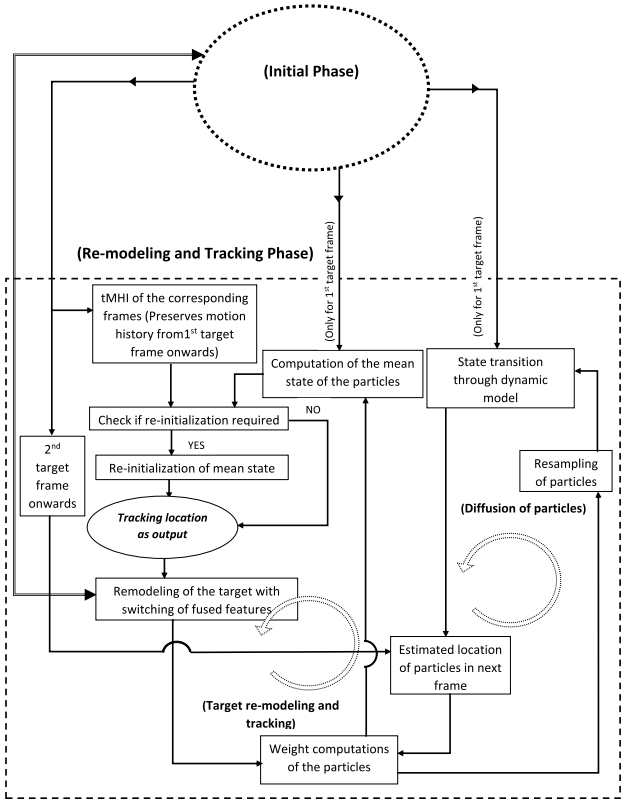

FIGURE 3. Block diagram of Re-modeling and Tracking phase of the proposed scheme.

3) DIFFUSION OF PARTICLES

The particles from the first target frame are passed through the state dynamic model for state transition as shown in Fig. 3. After state transition, the weights of the particles are computed, and the mean state of the particles is found out for the second target frame. To maintain the effective number of particles for the state transition of the next time step, resampling is carried out. The process consisting of remodeling, reinitialization, and the transition of the particle by the state dynamic system is repeated till all the frames are exhausted or in a real time set up this process is repeated to continuously track the target object of a real world scenario.

IV. TARGET MODELING USING DISCRIMINATING FEATURE FUSION

A. FEATURE SELECTION

Target modeling is one of the key aspects of tracking of the object. For efficient tracking of the object in the scene, the target should be differentiated from the background area in each frame. Hence, attempts have been made to model the target with features that will differentiate the target from the rest of the scene in a given frame. Therefore, initially features that possess the discriminating attribute are selected. The discriminating features are selected as follows. Fig. 4 shows the object with the template that encompasses the complete object and a very small portion of the background. But, the extended template is completely filled with background areas. Let f_1 be the first feature considered for both the template and the extended template. The feature is extracted from the template and extended template region. Thereafter the histogram of the featured object and the featured background are found that correspond to the foreground and background model respectively. It is to be noted that the template predominantly covers the object. Let $h_{o_{f1}}$ denotes

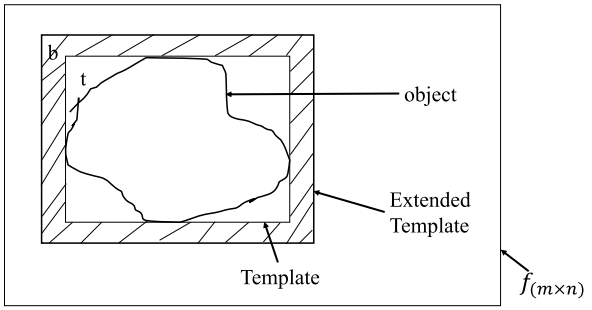

FIGURE 4. A frame $f(m \times n)$ with object, template and extended template regions. The shaded area is the background area included in the extended templated.

the histogram of the template region with the object and $h_{b_{f1}}$ denotes the histogram of the background area in the extended template region. If the above foreground and background distributions have a similarity of a high degree, then the chosen feature has the potential to model both background and foreground as well. Since we need to choose a feature that has the attribute to discriminate between background and foreground, we need to choose a feature whose corresponding background and foreground distributions are having less degree of similarity. The similarity measure between the two feature histograms $h_{o_{f1}}$ and $h_{b_{f1}}$ corresponding to f_1 feature is determined by Bhattacharyya's similarity measure [54] which is given by,

$$d = \sqrt{1 - \rho[h_o, h_b]}, \tag{1}$$

where d is the Bhattacharyya distance and $\rho[h_o, h_b]$ is the Bhattacharyya coefficient which is computed as;

$$\rho[h_o, h_b] = \sum_{u=1}^{m} \sqrt{(h_o^u, h_b^u)}, \tag{2}$$

where u represents the bins of the histogram and m is the maximum number of bins. $\rho[h_o, h_b]$ is an indicator of the similarity between the two distributions. Besides, h_o^u and h_b^u denote the u^{th} bin of the object and background histogram respectively. When the two distributions are same, the value of $\rho[h_o, h_b]$ is unity and the value of $\rho[h_o, h_b]$ is zero when the two distributions are orthogonal to each other. Let us denote γ to be the dissimilar coefficient between the object

and background distributions for a given feature $'f'$ where,

$$\gamma_f = 1 - \rho_f[h_o, h_b]. \tag{3}$$

If γ_f is greater than a threshold th_d, then the object and background distributions are dissimilar and hence the selected feature f has the potential of discriminating foreground and background. Following the above process, the discriminating features are selected for a given object position in the scene. In this work, three features namely, RGB color, LBP, and HOG are considered and the corresponding dissimilar coefficients are determined. The highest coefficient corresponds to the high degree of dissimilarity followed by others. In this process the discriminating ability of a given feature is ascertained and thereafter feature fusion takes place.

B. FEATURE FUSION

In order to improve the accuracy of target modeling, multiple discriminating features are fused probabilistically and the fused distribution is considered as the model of the target. In this work, we have chosen two discriminating features to be fused to obtain the fused distribution. In the template and extended template of Fig. 4, the two most discriminant features out of the three are selected. Let f_1 and f_2 denote the two discriminating features chosen over the object and let the corresponding histogram distributions are h_{f_1} and h_{f_2} respectively. These two distributions are shown in Fig. 5(a) and Fig. 5(b) respectively. As seen from these figures, the distributions corresponding to the two discriminating features are different. These two distributions are probabilistically fused to result in the fused distribution which is used to model the target and is shown in Fig. 5(c). The two histogram distributions are binwise fused. The two distributions are fused probabilistically as,

$$h_{f_d} = w_{d_1} h_{f_1} + w_{d_2} h_{f_2}, \tag{4}$$

where h_{f_d} denotes the fused distribution corresponding to the discriminant attributes. w_{d_1} and w_{d_2} denote the weights for the distributions of feature f_1 and feature f_2. These weights are determined as follows.

$$w_{d_1} = \frac{\gamma_{f_1}}{\gamma_{f_1} + \gamma_{f_2}}, \tag{5}$$

and

$$w_{d_2} = \frac{\gamma_{f_2}}{\gamma_{f_1} + \gamma_{f_2}}, \tag{6}$$

where γ_{f_1} and γ_{f_2} denote the dissimilar coefficients of feature f_1 and f_2 respectively.

Out of three features i.e., LBP, mean RGB color, and HOG, three sets of fused distributions are found. The fusion is carried out in both the template area and the extended template area. Out of three such fused distributions, the one that has the most discriminant attribute of discriminating the background and foreground is chosen for modeling the target.

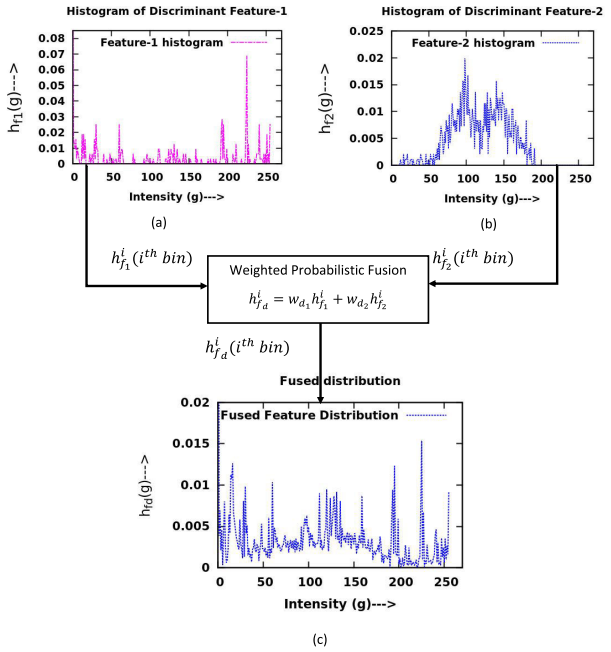

FIGURE 5. Probabilistic fusion of two distributions (a) histogram for 1^{st} feature, (b) histogram for 2^{nd} feature, (c) Fused feature histogram resulting from binwise fusion of 1^{st} and 2^{nd} feature.

V. REINITIALIZATION OF TARGET LOCATION

The target is tracked in every frame and this tracking is achieved by determining the mean state of the particles passed through the dynamic model. It may happen that the estimated mean state of a given frame may deviate from the expected one, which in turn deviates the tracker from the target position. If this deviation is not corrected after a certain degree then the percentage of deviation may increase and eventually the tracker may completely fail to track the object. This phenomenon is shown in Fig. 6. As seen from Fig. 6, the expected mean state encompasses the object entirely and hence has tracked the object accurately whereas the estimated mean state as shown with dotted lines covers only a portion of the target and hence tracker is deviated from the expected one. If this deviation is not corrected then eventually the tracker may fail completely to track the object. In order to ameliorate this situation, we introduce the concept of reinitialization of the mean state of the estimated particles of a frame if the tracker position has deviated beyond a threshold. This is achieved based on the following two notions; (*i*) Integrating the timed motion history information of the frame, (*ii*) Searching the neighborhood of the current tracker location to improve the accuracy of the tracking.

The timed motion history of a given frame is computed as follows. The motion history image (MHI) expresses the motion flow by using the intensity of every pixel in a temporal manner. A motion history of image at time $'t'$ for the pixel (x, y) is denoted as MHI_t which is computed as,

$$MHI_t(x, y) = \begin{cases} t, & \text{if } |D_t(x, y)| > \sigma \\ MHI_{t-1}(x, y), & otherwise. \end{cases} \tag{7}$$

where $D_t(x, y)$ contains the difference of images and σ is the difference threshold. Timed motion history image (*tMHI*) [52], [53] is an extension of *MHI*. In this notion, the motion history image (*MHI*) is generalized by directly encoding the actual time in a floating point format, which is called a timed-motion history image (*tMHI*). A history of temporal changes is stored at each pixel location which decays over time. The *tMHI* is not updated by the frame number but by the time stamp of the video sequence. The *tMHI* can be computed as,

$$tMHI_\sigma(x, y) = \begin{cases} \tau, & \text{if current silhouette at } (x, y) \\ 0, & \text{else if } tMHI_\sigma(x, y) < \tau - \delta, \end{cases} \tag{8}$$

where τ is the current time stamp and δ is the decay parameter that determines the maximum motion length. The time motion history indicates the number of moving pixels in the template. In the reinitialization phase, for every tracker

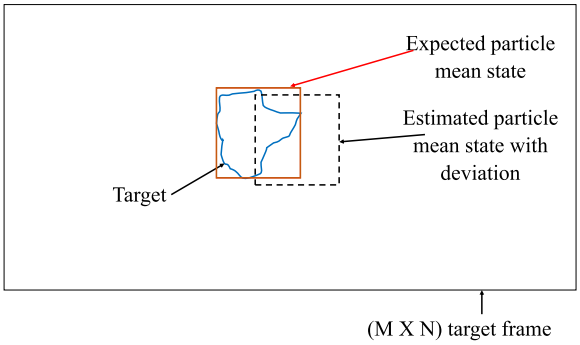

FIGURE 6. True mean state of the target and the estimated mean state with deviation in a given target frame.

position, possible tracker positions in the neighborhood are searched and thereafter the percentage of moving pixels encompassed by each neighborhood tracker is determined with the help of the *tMHI* of that frame. If x_i is the pixel having motion in a tracker of width W and height H and if,

$$\frac{1}{W \times H}\left(\sum_{i=1}^{W \times H} k_{x_i}\right) < th_t \tag{9}$$

then the tracker needs to be repositioned and hence the mean state needs to be reinitialized. In the above equation, k_{x_i} is unity when x_i is a motion pixel and zero otherwise. In order to reinitialize the tracker, the different possible positions of the tracker in the neighborhood in all directions are searched as shown in Fig. 7. In each of the neighborhood tracker position, the number of moving pixels is determined with the help of the timed motion history of the frame. If the percentage of the moving pixels of the target for a given tracker position is greater than that of a preselected threshold th_t, then it is assumed that the tracker covers the target substantially. As a thumb rule, we have assumed that if the tracker covers more than 50 percent of the target then reinitialization is not necessary and hence th_t is chosen as 0.5. Therefore, the tracker having the maximum number of

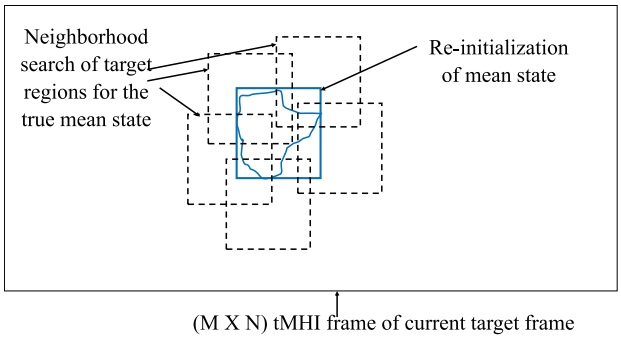

FIGURE 7. Neighborhood search space of the target frame when the estimated target location has motion information less than a threshold value. The motion information is obtained from the corresponding tMHI frame.

moving pixels is chosen from a neighborhood search. The mean state corresponding to this position becomes the new mean state of particles and hence becomes the new tracker position. This overcomes the problem of divergence of the tracker.

VI. REMODELING OF THE TARGET IN EACH FRAME

The scene condition changes in every frame and hence any single feature or a set of two features selected to model the target for the entire frame sequence may not be appropriate for tracking the object accurately over different frames of the video. The tracking accuracy of the tracker greatly depends upon the accurate modeling of the target in a given frame. Therefore, remodeling of the target is necessary in every frame. To have effective tracking of the object, fused feature distributions are found out which discriminate the foreground and background to the maximum possible extent. The most discriminating fused feature distribution is found by the notion of switching among the set of fused feature distributions. The fused feature distributions are found out and if a particular fused feature distribution is not able to achieve the required level of discrimination, then switching of the fused feature distribution happens to select the most discriminating fused features. In order to overcome the remodeling error that might happen due to the propagation of tracking error of the previous frame, it is desirable to have a high similarity of the remodeled target with that of the original target model. This ensures that the target model does not deviate from the original model and will supplement the tracking accuracy. The block diagrammatic representation of the entire process of remodeling the target is presented in Fig. 8. As seen from Fig. 8, there are three phases in the remodeling of the target. They are; (*i*) Modeling of the target region and the extended target region with fused features, (*ii*) Adaptation of the threshold of similarity, (*iii*) Decision for switching of features to remodel the target.

A. MODELING OF THE TARGET REGION AND THE EXTENDED TARGET REGION WITH FUSED FEATURES

First, for a given frame, the template with the object is chosen and an extended region of the template having a

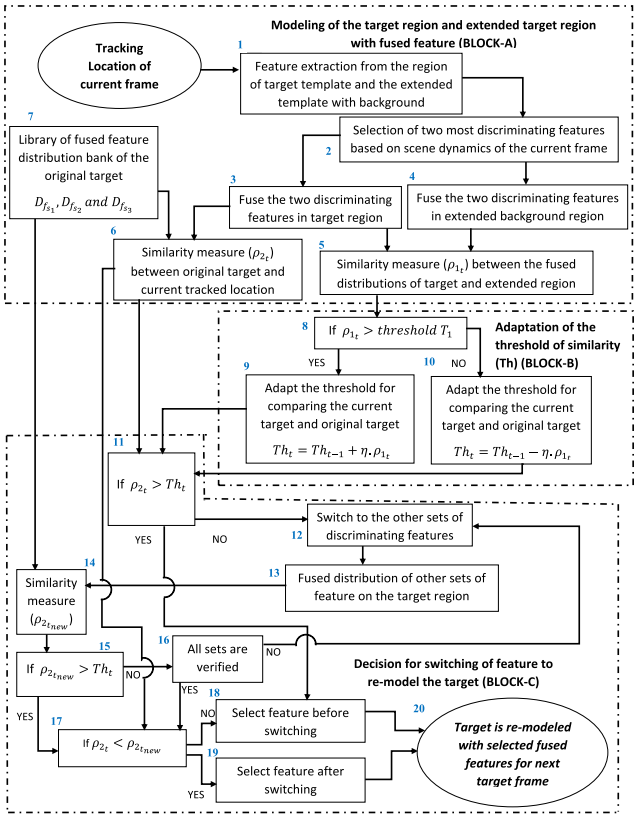

FIGURE 8. Remodeling of the target with the notion of switching of fused feature.

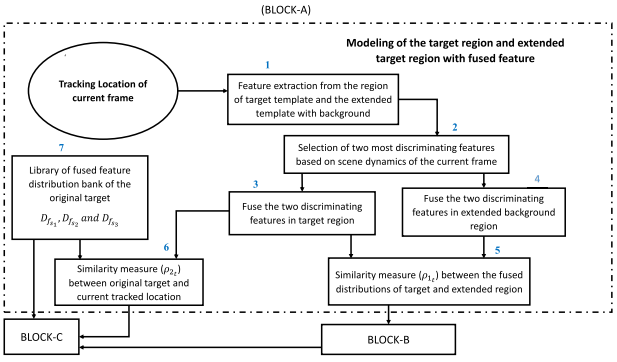

FIGURE 9. Block diagrammatic representation of the process of target modeling by fused feature distribution.

background is chosen as shown in Fig. 4. The features that have the attribute of discriminating the object from the background are selected. Out of these discriminating features, the two most discriminating features are chosen and fused probabilistically as presented in Sec. III-B. These two discriminating features are fused separately both in the target region and background regions as explained in Fig. 9. The fused distribution of the target region needs to be dissimilar with that of the background region so that the tracker will be able to differentiate the object and background.

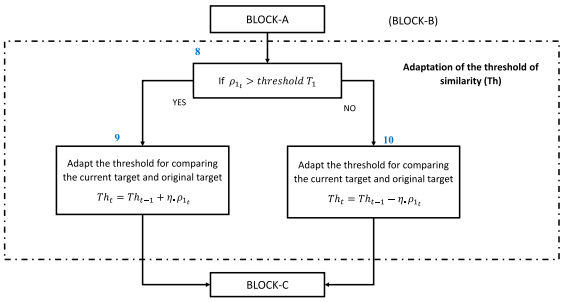

FIGURE 10. Adaptation of the threshold of similarity between target model of a given frame with that of the original target model.

B. ADAPTATION OF THE THRESHOLD OF SIMILARITY

The original target model may not be the appropriate model for all the frames because of the change in the scene conditions. Therefore, modeling of the target at each frame would be more appropriate to reduce the modeling error and improve the tracking accuracy. This is called as remodeling of the target. To model the target of a given frame effectively, the mean estimated tracking location is considered and the target is modeled by the fused feature distribution that has the most discriminating attribute to discriminate the background and foreground regions. But the estimated tracker position might be the deviated tracker position as compared to the expected tracker location. Therefore, the object template is expected to have some portion of the target and the rest as background. The rest portion of the target may be present in the extended template region. In such a scenario, if the discriminant fused feature distributions are chosen to model the target, the modeling may not be appropriate and hence modeling error will be reflected. This modeling error is expected to increase the tracking error, which may further increase the modeling error in the subsequent frames. This process eventually may lead to tracking failure. To overcome this problem, a discriminant fused feature based target model of a given frame is compared with the original target model. If there is less degree of similarity, then the chosen model will incur large modeling errors. Hence a different set of fused distributions needs to be chosen from the fused feature bank which will enhance the degree of similarity with that of the original target model. This is carried out by the notion of switching of features. This high degree of similarity will reduce the modeling error. Since the scene dynamics change from frame to frame, the threshold of similarity also needs to be adapted. Initially, a threshold $'Th'$ is chosen for similarity and is updated. As seen from Fig. 10, the similarity between the foreground and the extended background distributions at t^{th} time instant is determined by Bhattacharyya's coefficient ρ_{1_t}. If the ρ_{1_t} is greater than the chosen threshold T_1, this indicates that the chosen model results in more similarity between foreground and background. Hence with the use of this model, the tracker will have the tendency to deviate more towards the background region than the foreground region. This will deviate the tracker from the true mean state and hence incur more tracking errors. In such a scenario, even

though the discriminating ability of the model is less, the target modeling should be accurate and hence should have a high degree of similarity with the original target. To ensure this, the threshold $'Th'$ should be increased to ascertain the high degree of similarity with the original target. This will reduce the target modeling error and in turn the tracking error. Similarly, when the similarity ρ_{1_t} is less than the threshold T_1, the model has a discriminating ability and hence a lower value of threshold $'Th'$ for similarity with that of the original target will suffice for effective modeling. Hence, the proposed adaptation strategy for this is as follow;

$$Th = \begin{cases} Th_{t-1} + \eta\rho_{1_t}, & \text{if } \rho_{1_t} > T_1 \\ Th_{t-1} - \eta\rho_{1_t}, & \text{if } \rho_{1_t} < T_1. \end{cases} \tag{10}$$

where η is the control parameter and ρ_{1_t} is the similarity between the fused distribution in the target region and extended template background region. T_1 is the threshold to determine the dissimilarity between the target region and the background region of the extended template.

C. DECISION FOR SWITCHING OF FUSED FEATURE DISTRIBUTION TO REMODEL THE TARGET

It may so happen that due to various external factors such as illumination variation over the scene, the fused feature distribution chosen to discriminate the background and object may not be the appropriate features for target modeling. In order to improve model accuracy, different fused distributions may be used to model the target. Therefore, the chosen fused feature distribution of the target modeling in a given frame is compared with corresponding fused feature distributions from the feature bank having discriminating attributes. These distributions of the feature bank are different models of the original target corresponding to different features. Let $D_{f_{s_1}}$, $D_{f_{s_2}}$, and $D_{f_{s_3}}$ denote the three fused distributions of the feature bank corresponding to three sets of discriminating features of (f_1, f_2), (f_2, f_3), and (f_1, f_3). The distributions are arranged in descending order of discriminating attribute i.e.; $D_{f_{s_1}} > D_{f_{s_2}} > D_{f_{s_3}}$. As seen from the 3^{rd} block of Fig. 11, the chosen fused distribution D_{f_t} is compared with the corresponding target model from the feature bank $D_{f_{s_i}}$ and the Bhattacharyya similarity is denoted as ρ_{2_t}. If $\rho_{2_t} > Th_t$, then the discriminating feature selected for the t^{th} frame is accepted and the target is remodeled with the selected fused feature. But, if $\rho_{2_t} < Th_t$, this implies that the chosen fused distribution corresponding to the discriminating features is not appropriate for modeling and hence switches to a different set of discriminating features as shown in the 12^{th} block of Fig. 11. The fused distribution of the next set of fused feature distribution is denoted by $D_{f_{t_{new}}}$ and $D_{f_{t_{new}}}$ is compared with the corresponding fused distributions of the feature bank and this similarity is denoted as $\rho_{2_{t_{new}}}$. If $\rho_{2_{t_{new}}} > Th_t$, then the similarity value $\rho_{2_{t_{new}}}$ is compared with the earlier similarity value of ρ_{2_t}. If $\rho_{2_{t_{new}}}$ is greater than ρ_{2_t} then the switched feature distribution is considered for modeling and if $\rho_{2_{t_{new}}}$ is less than ρ_{2_t} then it implies that

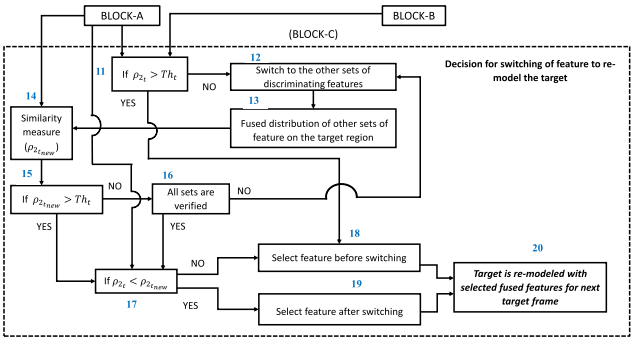

FIGURE 11. Remodeling of the target with appropriate fused features from the feature bank.

the switched fused distribution is not suitable for modeling. Thereafter it is checked in the 16^{th} block that whether all the fused distributions of the feature bank are exhausted. If it is not exhausted, switching takes place for the next fused distribution, and the process is repeated till we find a distribution whose similarity is greater than the threshold Th_t for modeling the target. In this process, the appropriate fused distribution is selected for remodeling the target at the t^{th} frame.

VII. TRACKING USING PARTICLE FILTER

The proposed particle filter based algorithm tracks the object in every frame using the notion of remodeling of the target at every frame and reinitialization of the tracker position as and when necessary. This is intended to track the posterior distribution of the object state. Initially, the object or the target is modeled by considering a rectangular template of width H_x and height H_y. This is represented by a particle $s = \{x, y, v_x, v_y, H_x, H_y\}$ where (x, y) are the coordinates of the location. v_x and v_y are the motion vectors in the x and y direction and H_x and H_y are the width and height of the rectangle respectively.

In initial frames, the object is encompassed by the tracker manually. Two features are chosen at a time from the set of Color, LBP, and HOG features, and the corresponding fused feature distribution is found out. Thus, three sets of fused feature distributions are determined to form the fused feature bank. For the first frame, a fused feature that has the potential to model both the target and the background is considered to model the target. A sample is drawn with a probability from the likelihood region of the target. Similarly, another sample is drawn from the second frame with a probability. Using both the samples a particle is generated and this particle is replicated N times to have a set of particles to be used for tracking. These particles are propagated through the following dynamic state model.

$$S_t = AS_{t-1} + W_t. \tag{11}$$

where S_t and S_{t-1} denote the set of particles at time step t, and $t-1$ respectively, and A denotes the state transition matrix. W_t denotes the Gaussian noise process at time t. In the tracking

phase, the fused feature distribution that can discriminate the foreground and background is considered. Hence, these sets of particles are modeled by the fused feature distributions having the most discriminating ability to differentiate the target and background. Each of the model distributions of these particles $p^{(n)}$ is compared with the initial target model D_{f_s} based on the similarity measure which is determined by the following Bhattacharyya coefficient [54];

$$d^{(n)} = \sqrt{1 - \rho[p^{(n)}, D_{f_s}]}, \tag{12}$$

where $d^{(n)}$ is the Bhattacharyya distance for n^{th} particle and $\rho[p^{(n)}, D_{f_s}]$ is the Bhattacharyya coefficient of n^{th} particle with that of the initial target model which is computed as;

$$\rho[p^{(n)}, D_{f_s}] = \sum_{u=1}^{m} \sqrt{(D_{f_s}^{u}, p^{(n)^u})}, \tag{13}$$

where u represents the bins of the histogram and m is the maximum number of bins. Each of the particles is assigned with a weight depending on the similarity measure, the more the similarity measures, the larger the weight, and vice versa. The mean state of these particles is the tracked position of the object in that frame which is given by;

$$E[S_t] = \sum_{n=1}^{N} \pi^{(n)} S_t^{(n)}, \tag{14}$$

where n is the number of particles. $S_t^{(n)}$ is the n^{th} sample particle weighted with corresponding sampling probability $\pi^{(n)}$. Thereafter mean estimated particle is checked for reinitialization as presented in Sec. 4. Thereafter, remodeling of the target in the current frame is achieved as presented in Sec. 5. After reinitialization and remodeling, the particles are again propagated through the dynamic model of (11) to the next frame for tracking the object. This process is repeated till all the frames for tracking are exhausted.

VIII. HARDWARE IMPLEMENTATION

The proposed scheme is intended to be implemented in real world scenario. Hence, the hardware implementation for real time setup needs to be evaluated. In this regard, a Raspberry Pi based hardware setup is developed as shown in Fig. 12 and the different components of the system are shown in Fig. 13. The specifications of the system are provided in Table 1. Input frames from different data sets are presented to the hardware in offline mode. Each frame is presented and the algorithm is executed in hardware and the tracked object is displayed on the monitor as shown in Fig. 12. Thus the frames are stored in the memory of the hardware and are input to the algorithm one by one. The tracked results are found to be similar to those obtained by simulation except the time taken by the hardware system is higher than that of simulation.

As observed from Table 6, the minimum average time of execution per frame in hardware is 1.7 *sec* in case of Kite-surf data sets with a tracker size of (40 × 20) pixels. Further it is observed that the average tracking time is dependent on the size of the tracker. Since the minimum time is found to be 1.7 *sec*, it is intuitively appealing that with enhanced features

Algorithm 1 Particle Filter Algorithm

Input 1: Video frame sequences with moving object. Initialize Target model and a set of particles $'s'$.

Output: Tracked Object frames.

1 : Select a particle set S_{t-1} at time $t-1$, having initial weights.

2 : Input the particles to the dynamic state model as per Eq. 11 to have N numbers of locations corresponding to particles at time t.

3 : Compute the discriminative fused feature distributions using Eq. 4 at every location of the set of particles.

4 : Use Bhattacharyya distance $'d'$ to compare the similarity between the fused feature distribution of a given location with that of the target model.

5 : Determine the weights of each particles as: $\pi_t^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} exp^{(-\frac{d^2}{2\sigma^2})}$, d is evaluated by Eq. 12.

6 : Estimate the mean state of the particles using Eq. 14 and check the condition for reinitialization of the mean state.

7 : Reinitialize of the mean state if necessary, otherwise go to step 8.

8 : Remodel the target at the mean state to be used for next frame.

9 : Resample the particles to maintain fixed number of particles N. Discard the particles with low weights and replicate the particles of high weights.

10 : Steps 2 to 9 are repeated for all the available frames.

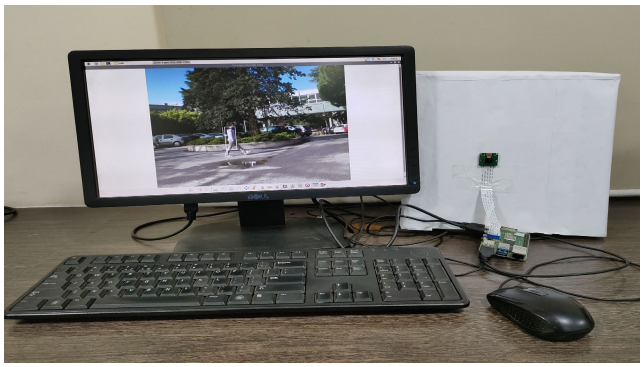

FIGURE 12. Raspberry Pi hardware along with the monitor for displaying the tracking in each frame.

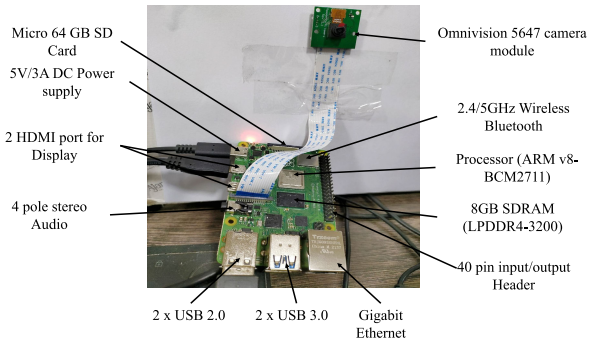

FIGURE 13. The detail components of Raspberry Pi system used for implementation of the proposed algorithm.

of the hardware, it will be a feasible candidate for real time implementation.

TABLE 1. Hardware specification.

Components	specification
Bord	Raspberry Pi 4 model B
Processor	Broadcom $BCM2711$, Quad core Cortex$-A72$ (ARM $v8$) 64-bit SoC @$1.5GHz$
RAM	8GB $LPDDR4-3200$ SDRAM
Wi-Fi and Bluetooth	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Ethernet	Gigabit Ethernet
USB Ports	2 USB 3.0 ports; 2 USB 2.0 ports
General Purpose I/O Pins (GPIO)	Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
Micro HDMI ports for Output	2× micro-HDMI ports (up to $4kp60$ supported)
The Display Serial Interface (DSI)	2-lane MIPI DSI display port
The Camera Serial Interface (CSI)	2-lane MIPI CSI camera port
Audio Video ports	4-pole stereo audio and composite video port
Video Decoder	$H.265$ ($4kp60$ decode), $H264$ ($1080p60$ decode, $1080p30$ encode)
Graphics	OpenGL ES 3.1, Vulkan 1.0
ROM and OS	$64GB$ micro-SD card with NOOB OS/ Raspberry Pi OS
Camera	$5MP$ Omnivision 5647 module, 2592×1944 resolution, supports $1080p@30fps$ or $720p@60fps$
Display device	1 LCD Display (DELL)
Power Supply	$5V$ DC via USB-C connector (minimum $3A*$)
Maximum Output from GPIO	$5V$ DC via GPIO header (minimum $3A*$)
Operating Temperature	$0-50$ degrees C ambient.

IX. EXPERIMENTAL CONDITION AND DATA SETS

In our experiment, the number of particles, N is chosen to be 100. The value of the threshold T_1 as mentioned in block 8 of Fig. 10 is chosen to be 0.94. We have considered a total of seven different videos, four from the following three data sets of LASIESTA [31], CDnet 2014 [29], and DAVIS 2016 [32], and three from OTB 100 [30]. Table 2 presents different data sets with their respective attributes. The coding developed for implementing the proposed algorithm is in $'C'$ programming language and the algorithm is processed in the Ubuntu platform of a machine having specifications, Intel (R) core (TM) $i5$ CPU, M480 @ 2.67GHz, with 4GB RAM.

TABLE 2. Data sets.

Name of the Data set	Name of the Database	Frame rate	Frame size	Typicality
Horse-jump -low	DAVIS 2016	$25fps$	180×320	Presence of moving shadows, and of Dynamic background entities.
walk (O_SU_01)	LASIESTA	$25fps$	288×352	Camouflage, Hard shadows, Reflection and Dynamic background
SnowFall	CDnet 2014	$24fps$	200×300	Bad weather and Dynamic entities in background
Kite-surf	DAVIS 2016	$25fps$	180×320	Dynamic entities in background
David3	OTB 100	$36fps$	300×400	Occlusion, Out-of plane-rotation and background clutter
Dog	OTB 100	$36fps$	240×352	Scale variation, Out-of plane-rotation and Deformation
Woman	OTB 100	$36fps$	288×352	Occlusion, Out-of plane-rotation, Illumination variation, Motion blur, Scale variation and Deformation

X. RESULTS AND DISCUSSION

Different results obtained for the seven different videos from different data base including OTB 100 [30] are presented in this section. However, the visual display of the tracking is presented only for four frames of Horse-jump-low and walk videos which are denoted as video 1 and video 2. However, we have presented the tracking error and trajectory of tracking for all four videos. Besides, for performance analysis of all the seven videos, we have presented the Average Overlap Scores (AOS), Success Scores (SS) and the Error Scores (ES). Besides, the diffusion of distributions of the particles for test video 1 are presented in Fig. 14 and Fig. 15. Additionally, the effect of remodeling and feature switching on tracking error is demonstrated in Fig. 17 for the walk video. Analogously, the effect of reinitialization on the trajectory is demonstrated for the Horse-jump-low video.

A. VISUAL PRESENTATION OF TRACKING

Though we have successfully tested the proposed scheme on seven different challenging videos from four data sets, tracking results in two videos are presented for validating visually. Four frames in each of the videos are considered to demonstrate the efficacy of the proposed method. Fig. 16 shows results on four selected frames of video 1, i.e the Horse-jump-low video. In this video, 22 frames are considered from 38^{th} to 59^{th} frames of video. Results for the 42^{nd}, 47^{th}, 55^{th}, and 59^{th} frames are presented in Fig. 16. These frames are chosen specifically to show the deviation of the tracker from the target object by different schemes and the potentiality of the proposed scheme to track the object. As observed from Fig. 16, the tracker in the Color model [34] deviates in the 42^{nd} frame and again completely failed in the 55^{th} and 59^{th} frames. But in case of the DC model [8] and the EOH-HSV model [7], the tracker is found to deviate in the 55^{th} and 59^{th} frames and thereafter deviated more. The

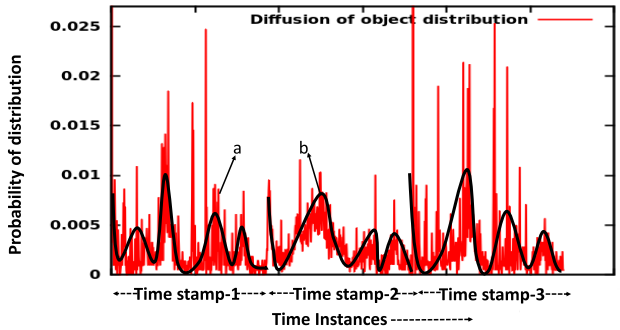

FIGURE 14. Diffusion of object distribution. (a) Diffusion of distribution in discrete domain, (b) The corresponding one in continuous domain.

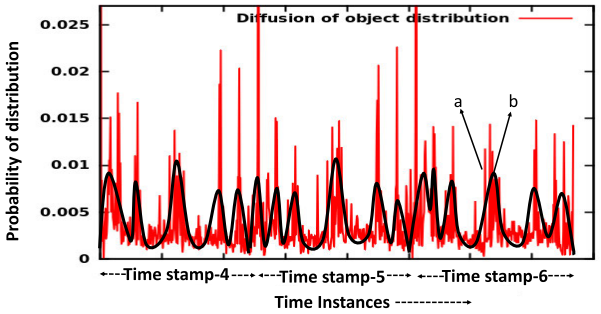

FIGURE 15. Diffusion of object distribution. (a) Diffusion of distribution in discrete domain, (b) The corresponding one in continuous domain.

tracker is found to deviate in the 59^{th} frame for the FFUTP model, our previously developed model [6]. But, the tracker in case of the proposed scheme could encompass the object well and thus could track the object even in complex scenes. In this video, the scene condition changes with the gallop of the horse and hence our proposed notion of remodeling by feature switching and reinitialization of tracker position could take care of the change in scene dynamics and track the object with minimum tracking error. Further, for the proposed algorithm there is no diverging trend of error in the subsequent frames.

The second illustration is presented in Fig. 23 with four frames from the second video i.e. walk video. As observed from Fig. 23, the person walks with change in illumination condition. As seen in Fig. 23, the tracker in the Color model failed in the 147^{th} frame and thereafter completely failed in the 171^{th} and 183^{rd} frames. The tracker in the DC model also showed a trend of deviation and completely deviated in the 171^{th} and 183^{rd} frames. Further the tracker in the case of EOH-HSV model deviated in the 171^{th} and 183^{rd} frame while the tracker in the FFUTP model is out of the object in 171^{th} and 183^{rd} frames. But, for the proposed model, the tracker exhibited deviating trend in 171^{th} frame but could again track the object completely in 183^{rd} frame. This is due to the special attributes of the proposed scheme.

B. DIFFUSION OF TARGET DISTRIBUTION

In the particle filtering approach, the target is modeled by the distribution of the particles. As the target moves over time, the distribution is expected to diffuse in order to track the object over different frames of the video. This phenomenon is demonstrated in Fig. 14 and Fig. 15. For the sake of illustration, the process of diffusion is demonstrated for video 1. Both these figures together show for six time stamps and as observed from these figures, the target distribution in one time stamp diffuses to the subsequent one thus enabling the tracker to track the object over different frames. In these figures, one time stamp corresponds to one frame. It is to be noted that the dominant peak of the distribution in one time stamp corresponds to the tracked target position. The particles move through the dynamic model resulting in the diffusion of the target distribution from one frame to the other. Fig. 14 shows the diffusion of the distribution for time stamps 1 to 3 and Fig. 15 shows the diffusion of the distribution for time stamps 4 to 6. This indicates that the proposed particle filter based scheme could track the object in different frames.

C. EFFECT OF FEATURE SWITCHING IN REMODELING AND REINITIALIZATION OF THE MEAN STATE OF THE PARTICLE

In the proposed particle filter based scheme the target is remodeled in every frame to take care of the change in scene dynamics. This takes place through the proposed notion of switching of fused feature distributions. Tracking error is affected by feature switching, remodeling and reinitialization and their effect on tracking error is shown in Fig. 17. For the sake of illustration, the effect of remodeling for test video 2 is shown in Fig. 17. As observed from this figure, the tracking error is reduced in each frame with remodeling. But even with remodeling, the tracking error increases and shows a diverging trend. In order to overcome this effect, the notion of a reinitialization of the mean state of the particle is combined with remodeling, and the error due to the combined effect is shown by curve $'C'$ of Fig. 17. As observed, the tracking error has reduced substantially and there is no diverging trend. Hence, the tracker could track the object even with the changing scene condition. Besides, the effect of the proposed notion of reinitialization on the trajectory of the tracker is shown in Fig. 18. This figure shows the original trajectory of the object and the trajectory of the tracker by the proposed model. This trajectory is obtained for the second testing video. As observed from Fig. 18, the coordinate of the tracker position for the 9^{th} frame is $(177, 71)$ which is away from the original trajectory coordinate of $(179, 66)$. Reinitialization of the tracker position brings the tracker position to $(189, 65)$ which is close to the original trajectory point of $(187, 63)$. It is to be noted that without reinitialization, the tracker position would have deviated more from the original trajectory and eventually might have completely deviated away from the trajectory. Similar observations are also made for other frames and hence the estimated tracker positions try to follow the original trajectory. Thus the

FIGURE 16. Testing video-1 (frame-42, frame-47, frame-55, frame-59). The original frames and tracked frames for different models and the proposed model.

reinitialization together with the remodeling helps the tracker to track the object in different frames with varying scene conditions.

D. TRACKING ERROR AND TRAJECTORY OF TRACKING

In order to analyze the results quantitatively in all the frames, tracking errors of different frames for different videos are plotted. The tracking error in a given frame is defined by the Euclidean distance expressed as; $\| m_{s_t} - m_{s_g} \|$, where m_{s_t} denotes the center coordinate of the tracker and m_{s_g} denotes the center coordinate of the ground truth. Besides error, the trajectory of tracking is plotted together with the original trajectory to show how close is the tracker's trajectory with that of the original trajectory. In the trajectory plot, the X and Y coordinates of the tracker at different frames are plotted to demonstrate tracking ability for the given scene. The errors and trajectories are also plotted for different models. Fig. 19 and Fig. 20 show the tracking error and trajectory plots respectively for video 1. As observed from Fig. 19, the error for the Color model diverges at the 55^{th} frame thus indicating the fact that the tracker has completely failed to track the object. This is quite evident from the frames presented in Fig. 16. Similar observations are also made in the case of the trajectory as presented in Fig. 20. As observed, in the case of the Color model, there is a large deviation from the original trajectory and hence the Color model failed to track the object. As observed from Fig. 20, the trajectories for other models are deviating from the original trajectory whereas the trajectory for the proposed model follows the original trajectory even towards the last frame of the video. These observations are also corroborated by the error plot presented in Fig. 19. It is also seen from the Fig. 19 that the errors for other models showed diverging trend towards last part of tracking. Further, it may also be

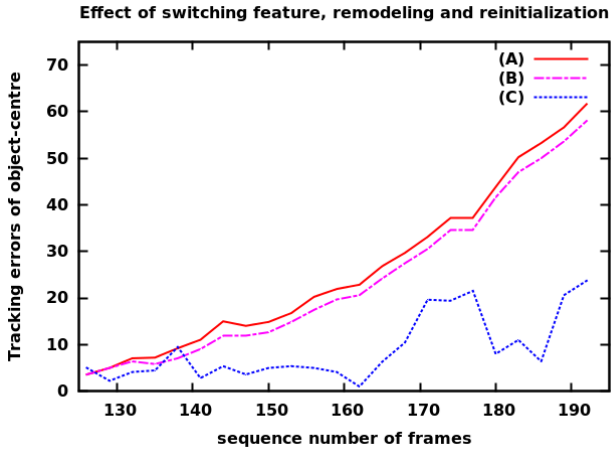

FIGURE 17. Effect of switching feature,remodeling and reinitialization (A) Fused feature modeling without remodeling for other frames, (B) Remodeling and feature switching, (C) Proposed method with remodeling, feature switching and reinitialization.

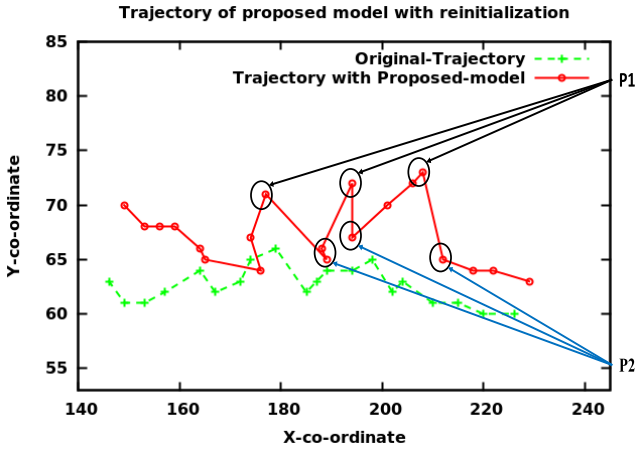

FIGURE 18. Demonstration of deviation of the mean position necessiating reinitialization (P1) and the mean position after reinitialization (P2).

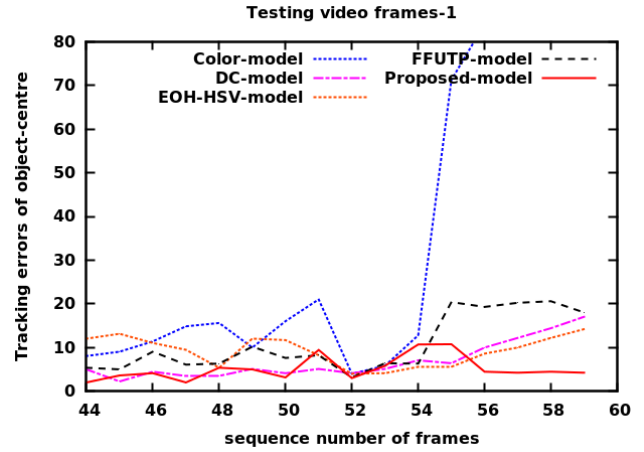

FIGURE 19. Error plots corresponding to different models for frame sequence of video-1.

observed that the tracking error for the proposed model is not only minimum but also has a converging trend towards the

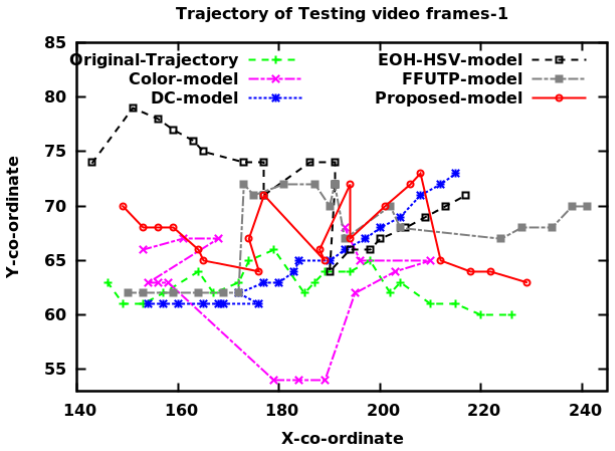

FIGURE 20. Trajectories of the mean tracker position obtained by different models and the original trajectory for video-1.

last part of tracking.This is due to the fused feature switching and reinitialization attributes of the proposed model. The effect of reinitialization is reflected in Fig. 18. The errors and trajectory plots for video 2 are presented in Fig. 21 and Fig. 22. As observed from Fig. 21, the error in the Color model diverges whereas the error plots of the EOH-HSV model and the proposed model are close to each other. But the errors in case of the proposed model in most of the frames are less than those of the EOH-HSV model. Further, the result of the two models such as the DC model and the FFUTP model show diverging trend after the 173^{rd} frame. These observations are also reflected in the trajectories of different models shown in Fig. 22. As observed from Fig. 22, the trajectories of the proposed model, and that of EOH-HSV model are found to follow the original trajectory. The trajectory of the Color model diverges after a few frames whereas the trajectory of the DC model shows the diverging trend. But the FFUTP model follows the original trajectory for few frames and thereafter diverges. The proposed model could follow the original trajectory throughout the tracking process because of the combined effect of remodeling and reinitialization. The average errors for different videos are presented in the Table 3, where it is observed that average errors for videos are minimum ones for the proposed model.

The proposed algorithm is also tested for video 3 consisting of snowfall data and errors and trajectory plots are shown in Fig. 24 and Fig. 25 respectively. As observed from Fig. 24, the DC model like the previous two examples diverged. The error plots for the DC model and FFUTP model also diverged after a few frames. But the EOH-HSV model which exhibited appreciable tracking capability for the last two examples failed to track the object. Hence, large errors with the diverging trend are observed in the last few frames. But the proposed model incurred less error with a converging trend. These observations are also reflected in trajectory tracking as shown in Fig. 25. It may be observed that the trajectory of the

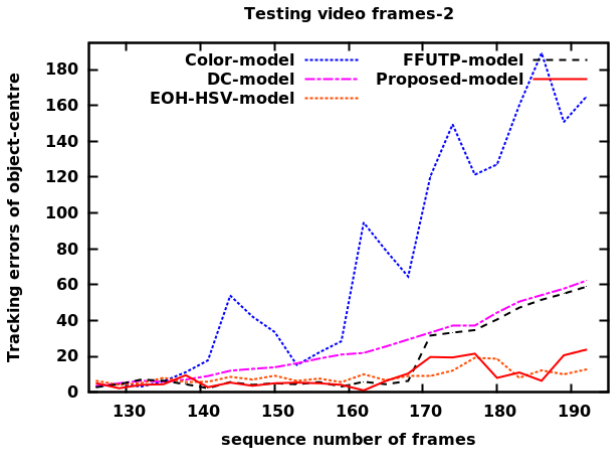

FIGURE 21. Error plots corresponding to different models for frame sequence of video-2.

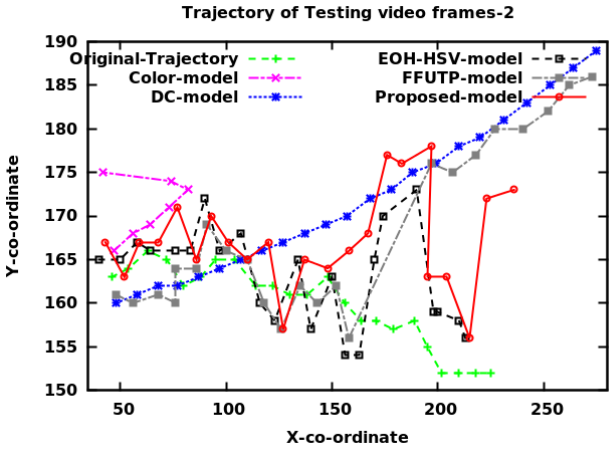

FIGURE 22. Trajectories of the mean tracker position obtained by different models and the original trajectory for video-2.

proposed model is closely following the original trajectory. Thus, the proposed model worked well in this bad weather data set.

The fourth example considered is the Kite-surf video from the DAVIS-2016 database and the error and the trajectory plots are presented in Fig. 26 and Fig. 27 respectively. In this case, also error plot of the EOH-HSV model is close to our proposed model while the errors for the rest of the models diverged after a few frames. This observation is reflected in trajectory tracking shown in Fig. 27. It may be observed that the trajectories of the proposed model and the EOH-HSV model are close to that of the original trajectory while the trajectories for the rest of the models are away from the original trajectory. In both these data sets the proposed model resulted in minimum average errors. Thus in all the four examples, the proposed model could track the object in all the videos with varying scene conditions.

E. FURTHER QUANTITATIVE ANALYSIS AND RESULTS ON OTB 100 DATA SETS

The efficacy of our proposed algorithim is demonstrated by evaluating the following quantitative measures on the four videos of different datasets and 3 videos from the object tracking benchmark data sets ($OTB100$) [30]. For the sake of illustration, two frames with trackers and the corresponding original frames of ($OTB100$) dataset are shown in Fig. 28. As observed from Fig. 28, the tracker does not deviate from the target and hence tracks the target object successfully. The center location errors on these ($OTB100$) videos are determined by our proposed tracking algorithim and are compared with the state-of-the-art algorithims. Let A_t^{GT} denotes the annotated area (i.e the number of pixels) of the object in the ground truth (GT) while A_t^T denotes the output area of the visual tracker at time step t. The error score S_t^{Err} at time step t is defined as the distance between the centers of the $GT's$ area and the output area of the tracker [55]. Considering the distance as L_p norm, S_t^{Err} can be expressed as,

$$S_t^{Err} = \sqrt[p]{\|Center(A_t^{GT}) - Center(A_t^T)\|^p}. \tag{15}$$

In our evaluation, we have taken p value as 2. The center location errors are averaged over number of frames with positive templates (error less than a threshold) to find the mean of the center location errors. Center location errors are evaluated in terms of pixels and hence smaller error indicates better tracking accuracy and vice versa. The center location errors of the proposed algorithim on the ($OTB100$) data sets are presented in Table 4 together with those of the state-of-the-art algorithims [56]. The errors are in pixels and the best results are presented in bold faces. As observed, for the Dog video, the average error is 3.6 which is very close to the best result of 3.5. In case of other two videos, the mean errors are less than many state-of the-art algorithims. The quantitative performance is also measured by Average Overlap Score (AOS), Success score (SS) and the Error Score (ES) [55]. The overlap score (OS) is defined as,

$$S_t^{Ov} = \frac{|A_t^{GT} \cap A_t^T|}{|A_t^{GT} \cup A_t^T|}. \tag{16}$$

It is to be noted that larger overlap score implies better tracking accuracy and vice versa. In an ideal case the overlap score should be unity. The success score is defined over the entire sequence of T frames which is defined as,

$$S^{Succ} = \frac{Success}{T}, \tag{17}$$

where $'Success'$ for a frame is defined as the output of the visual tracker which is considered to be correct based on the overlap score. A score is considered to be a $'Success'$ if it is greater than a predefined threshold. In our simulation, we have considered threshold to be 0.5 i.e when half of the object area is overlapped by the visual tracker's output area.

FIGURE 23. Testing video-2 (frame-129, frame-147, frame-171, frame-183). The original frames and tracked frames for different models and the proposed model.

TABLE 3. Mean of the errors of the trackers for different models with variety of datasets.

	Data sets							
Model Name	**Horse-jump -low**		**walk**		**Snow Fall**		**Kite -surf**	
	error	*std*	error	*std*	error	*std*	error	*std*
Color [34]	29.268	32.66	72.3	62.56	46.87	32.02	9.08	2.51
DISC [8]	7.04	3.83	25.486	18.415	23.22	25.54	17.51	7.67
EOH-HSV [7]	10.47	4.11	9.001	**3.88**	42.2	24.11	3.176	1.536
FFUTP [6]	9.78	6.1	18.478	20.064	25.08	35.9	14.4	3.35
Proposed	**5.6**	**2.66**	**8.903**	6.978	**16.8**	**5.33**	**2.47**	**0.8**

Denoting this threshold as T^{Acc}, the success score of (17) is defined as,

$$S^{Succ}_{T^{Acc}} = \frac{|t : S^{Acc}_t > T^{Acc}|}{T}. \tag{18}$$

where $'t'$ denotes the time step. Ideally the success score should be unity. But higher success score with respect to the threshold T^{Acc} indicates better tracking. Similarly, to compute the error score the threshold T^{Err} is fixed to be 20 pixels. The average error score is evaluated over those

TABLE 4. The center location errors of the state-of-the-art Algorithims and our algorithim.

Data sets	VTD	Stuck	TLD	CT	KCF	LRT	SH	TSP	MSP	Chaotic	ACSPF [56]	Our model
David3	66.7	107	–	88.7	4.3	86.0	13.1	4.1	7.6	3.7	**3.0**	10.03
Woman	137	4.3	–	113	10.1	157.3	4.3	8.0	10.5	5.7	**4.0**	7.1
Dog	11.0	5.7	4.2	7.0	4.2	3.7	**3.5**	8.5	9.3	**3.5**	**3.5**	3.6

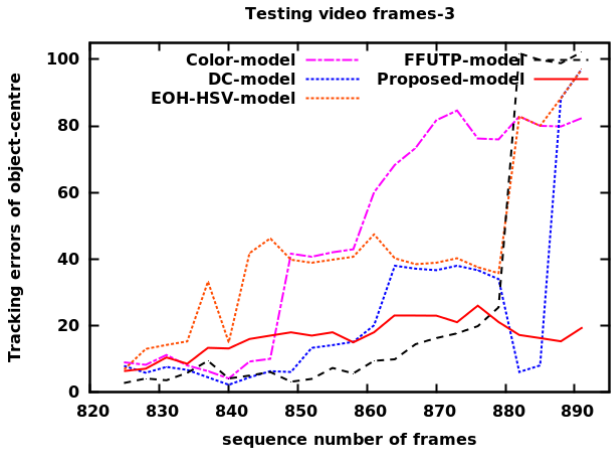

FIGURE 24. Error plots corresponding to different models for frame sequence of video-3.

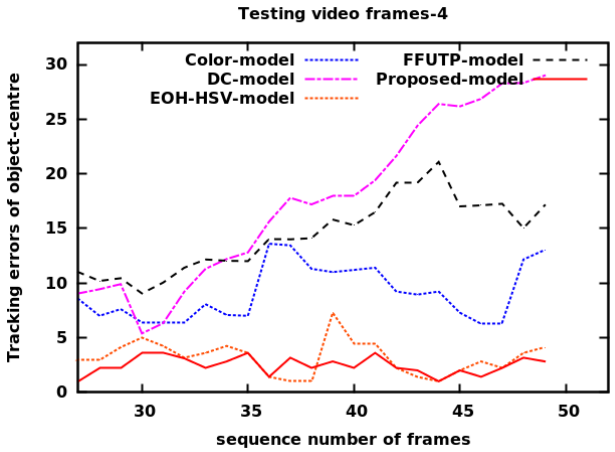

FIGURE 26. Error plots corresponding to different models for frame sequence of video-4.

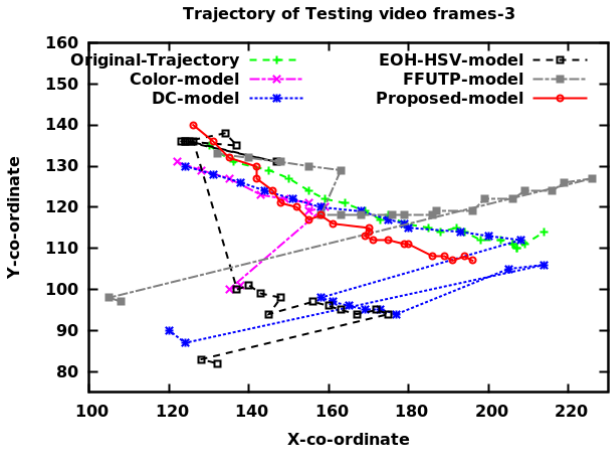

FIGURE 25. Trajectories of the mean tracker position obtained by different models and the original trajectory for video-3.

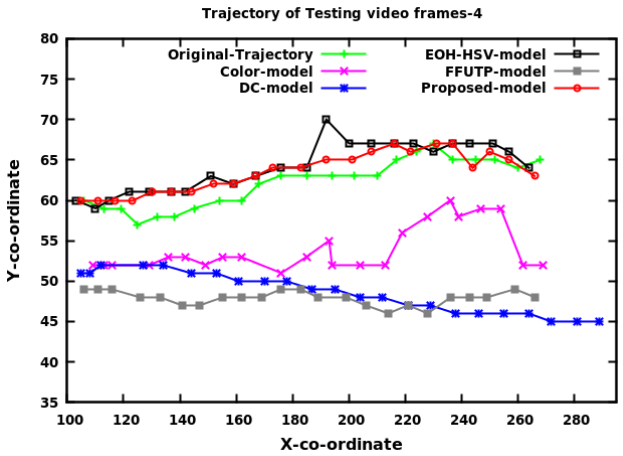

FIGURE 27. Trajectories of the mean tracker position obtained by different models and the original trajectory for video-4.

T frames, where the error in every frame is less than the threshold. The success for error threshold is defined as,

$$S_{T^{Err}}^{Succ} = \frac{|t : S_t^{Err} < T^{Err}|}{T}. \tag{19}$$

The success score for the error threshold T^{Err} is also to be high for high tracking accuracy and vice versa. Ideally, this success score for the error threshold should be unity. These quantitative measures for our proposed algorithim for all the videos including the three from (*OTB*100) data set are evaluated and are presented in Table 5. It is observed from this table that the success score is high for all the videos except

TABLE 5. Performance analysis of the proposed method in different data sets.

Data sets	Average overlap score (AOS)	Success score (SS)	Error score ES
Horse-jump-low	0.918	1.0	0.0
walk	0.877	1.0	0.12
SnowFall	0.996	1.0	0.24
Kite-surf	0.792	0.956	0.0
David3	0.737	0.850	0.190
Dog	0.921	1.0	0.0
Woman	0.653	0.772	0.375

Woman video. Similarly, the error score is as low as zero for three videos including the Dog video from *OTB*100 data

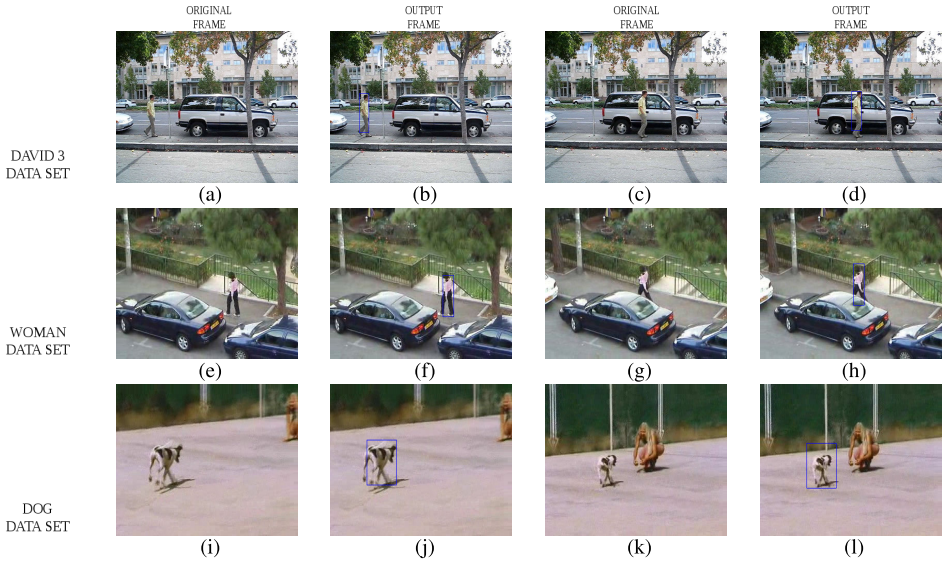

FIGURE 28. Tracking result of OTB-100 dataset (a) 7^{th} and (c) 64^{th} frame of David3 dataset, (b) and (d) represent corresponding tracking output, (e) 261^{th} and (g) 289^{th} frame of Woman dataset, (f) and (h) represent corresponding tracking output, (i) 13^{th} and (k) 64^{th} frame of Dog dataset, (j) and (l) represent corresponding tracking output.

set. The rest error scores are low except the Woman video. Further, it is observed that the *AOS* score is high for all the videos except the Woman video. Therefore, the proposed algorithims performs well in almost all the videos.

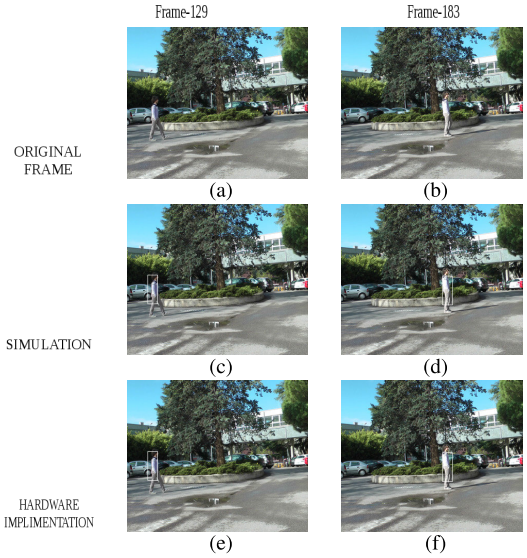

FIGURE 29. Comparison between simulation results obtained and by hardware implementation for Testing video-2 (frame-129, frame-183).

F. HARDWARE IMPLEMENTATION RESULT

We have tested all the examples in our Raspberry Pi processor based hardware setup and it is observed that the results obtained are the same as that of the simulation except higher execution time. Therefore, for the sake of illustration, we have

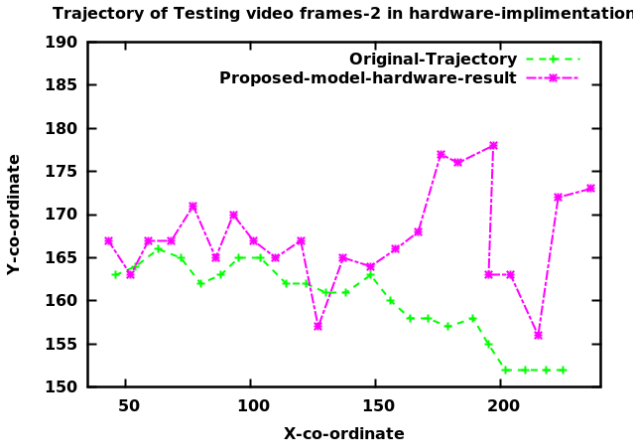

FIGURE 30. Trajectories of the mean tracker position obtained by proposed model and the original trajectory for video-2 in hardware implementation.

presented the trajectory of the tracker of video 2 in Fig. 30. The observations obtained for other models are similar to that of the simulation, and hence we have presented the trajectory of the proposed model which is found to follow the original trajectory. Fig. 31 presents the results obtained by simulation and it may be observed that the results of Fig. 30 and Fig. 31 are identical. The average time of execution for different examples are presented in Table 6. As observed from Table 6, the computational time with the hardware setup increases with the increase in size of the tracker. But, it may also be observed that with the increase in the size of the tracker, the execution time in hardware implementation is almost

TABLE 6. Average time of tracking per frame for the proposed method in different videos.

Data sets	Frame size	Particle size in pixels	Avg. time per frame in sec. in simulation	Avg. time per frame in sec. in hardware implimentation
Horse-jump-low	180 × 320	60 × 20	1.17	2.464
walk	288 × 352	60 × 20	1.227	2.396
SnowFall	200 × 300	116 × 68	6.726	16.038
Kite-surf	180 × 320	40 × 20	0.966	1.701

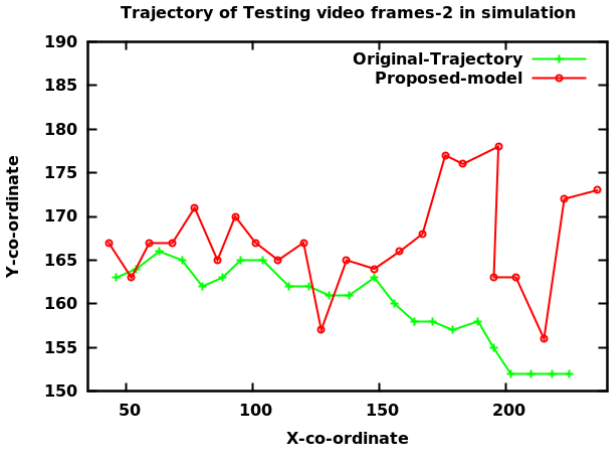

FIGURE 31. Trajectories of the mean tracker position obtained by proposed model and the original trajectory for video-2 in simulation.

twice than that of simulation. This is due to the configuration of the hardware setup. It may be inferred that with further advanced features of the hardware the execution time may further be reduced thus making it a possible candidate for real time implementation. It is found that the results obtained by hardware with different videos are same as those of the simulation. Therefore, the performance metrics such as Average Overlap Score (AOS) and Success Score (SS) of the results obtained by hardware setup are same as those of the simulation as presented in Table 5.

XI. CONCLUSION

In this research, a particle filter based video tracking scheme is proposed to achieve high tracking accuracy in real world scenarios with illumination variation, bad weather conditions, shadows, and dynamic entities of the background. The tracker usually deviates and eventually fails after a few frames if the initial target model remains the same for the entire tracking phase. In order to overcome the tracking failure, the target is remodeled in each frame of the video. To have an effective model, the target is modeled by fused feature distributions where the features are chosen based on the discriminating ability of the feature to differentiate the target and background. A fused feature distribution bank is created and the most appropriate fused feature distribution is selected. This modeling improved the tracking capability of the tracker but the tracker is found to deviate under certain complex scenes which may be due to the combined effect of the above mentioned scene conditions.

In order to overcome this problem, the notion of a reinitialization of the tracker position is proposed based on the concept of the time motion history of the object. It is found that the combined effect of remodeling and reinitialization could track the object in different complex scenarios with improved tracking accuracy. To make the proposed scheme a potential candidate for real time implementation, the proposed scheme is successfully implemented in Raspberry Pi based hardware system. The tracking accuracy is found to be similar to that of simulation but the only difference is that the computational time is more in hardware than that of simulation. Hence, with further enhanced hardware features, the scheme can be implemented in real time in a real world scenario. The scheme is successfully tested with different challenging videos from the OTB 100, DAVIS 2016, LASIESTA, and CDnet 2014 data sets. In most of the cases, the performance of the proposed scheme is found to be superior to the existing schemes.

REFERENCES

[1] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Computer Vision—ECCV '96*, B. Buxton and R. Cipolla, Eds., Berlin, Germany: Springer, 1996, pp. 343–356.

[2] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Radar Signal Process.*, vol. 140, no. 2, p. 107, 1993.

[4] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice* (Statistics for Engineering and Information Science). New York, NY, USA: Springer, 2001, doi: 10.1007/978-1-4757-3437-9.

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

[6] J. Panda and P. K. Nanda, "Particle filter-based video object tracking using feature fusion in template partitions," *Vis. Comput.*, vol. 39, no. 7, pp. 2757–2779, Jul. 2023.

[7] Z. Li, S. Gao, and K. Nai, "Robust object tracking based on adaptive templates matching via the fusion of multiple features," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 1–20, Apr. 2017.

[8] R. Huan, S. Bao, C. Wang, and Y. Pan, "Anti-occlusion particle filter object-tracking method based on feature fusion," *IET Image Process.*, vol. 12, no. 9, pp. 1519–1530, 2018.

[9] Z. Li, S. He, and M. Hashem, "Robust object tracking via multi-feature adaptive fusion based on stability: Contrast analysis," *Vis. Comput.*, vol. 31, no. 10, pp. 1319–1337, Oct. 2015.

[10] J. Panda and P. K. Nanda, "Video object-tracking using particle filtering and feature fusion," in *Proc. Advances in Electrical Control and Signal Systems*, vol. 665, G. Pradhan, G. Morris, and N. Nayak, Eds., Singapore: Springer, 2020, pp. 945–957.

[11] D. Roller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.

[12] E. Polat, M. Yeasin, and R. Sharma, "A 2D/3D model-based object tracking framework," *Pattern Recognit.*, vol. 36, no. 9, pp. 2127–2141, Sep. 2003.
[13] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1331–1342, Dec. 2011.
[14] Y. Fang, C. Wang, W. Yao, X. Zhao, H. Zhao, and H. Zha, "On-road vehicle tracking using part-based particle filter," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4538–4552, Dec. 2019.
[15] J. Choi and M. Maurer, "Local volumetric hybrid-map-based simultaneous localization and mapping with moving object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2440–2455, Sep. 2016.
[16] J. Wang, S. Xuan, H. Zhang, and X. Qin, "Target tracking method based on adaptive structured sparse representation with attention," *IEEE Access*, vol. 8, pp. 79419–79427, 2020.
[17] N. Morales, J. Toledo, L. Acosta, and J. Sánchez-Medina, "A combined voxel and particle filter-based approach for fast obstacle detection and tracking in automotive applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1824–1834, Jul. 2017.
[18] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 1158–1163.
[19] Y. Zhong, A. K. Jain, and M.-P. Dubuisson-Jolly, "Object tracking using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 544–549, May 2000.
[20] H. Tehrani Niknejad, A. Takeuchi, S. Mita, and D. McAllester, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 748–758, Jun. 2012.
[21] M. A. Greminger and B. J. Nelson, "Deformable object tracking using the boundary element method," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2003, pp. 289–294.
[22] F. Gu, J. Lu, and C. Cai, "RPformer: A robust parallel transformer for visual tracking in complex scenes," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
[23] F. Gu, J. Lu, C. Cai, Q. Zhu, and Z. Ju, "EANTrack: An efficient attention network for visual tracking," *IEEE Trans. Autom. Sci. Eng.*, early access, Oct. 3, 2004, doi: 10.1109/TASE.2023.3319676.
[24] D. Yuan, X. Shu, Q. Liu, and Z. He, "Aligned spatial–temporal memory network for thermal infrared target tracking," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 3, pp. 1224–1228, Mar. 2023.
[25] F. Gu, J. Lu, C. Cai, Q. Zhu, and Z. Ju, "VTST: Efficient visual tracking with a stereoscopic transformer," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 3, pp. 2401–2416, Jun. 2024.
[26] X. Zhao, L. Lyu, J. Zhang, and C. Lyu, "An image-constrained particle filter for 3D human motion tracking," *IEEE Access*, vol. 7, pp. 10294–10307, 2019.
[27] S. Munder, C. Schnorr, and D. M. Gavrila, "Pedestrian detection and tracking using a mixture of view-based shape–texture models," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 333–343, Jun. 2008.
[28] M. Younsi, M. Diaf, and P. Siarry, "Automatic multiple moving humans detection and tracking in image sequences taken from a stationary thermal infrared camera," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113171.
[29] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.
[30] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[31] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA," *Comput. Vis. Image Understand.*, vol. 152, pp. 103–117, Nov. 2016.
[32] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.
[33] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Computer Vision—ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., Berlin, Germany: Springer, 2002, pp. 661–675.
[34] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.
[35] S. S. Moghaddasi and N. Faraji, "A hybrid algorithm based on particle filter and genetic algorithm for target tracking," *Expert Syst. Appl.*, vol. 147, Jun. 2020, Art. no. 113188.
[36] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 212–219.
[37] C. Chen and D. Schonfeld, "A particle filtering framework for joint video tracking and pose estimation," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1625–1634, Jun. 2010.
[38] N. Bouaynaya and D. Schonfeld, "On the optimality of motion-based particle filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1068–1072, Jul. 2009.
[39] X. Gao, G. Xu, S. Li, Y. Wu, E. Dancigs, and J. Du, "Particle filter-based prediction for anomaly detection in automatic surveillance," *IEEE Access*, vol. 7, pp. 107550–107559, 2019.
[40] D. Y. Kim, E. Yang, M. Jeon, and V. Shin, "Real-time level set based tracking with appearance model using rao-blackwellized particle filter," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 4125–4128.
[41] M. D. Jenkins, P. Barrie, T. Buggy, and G. Morison, "Selective sampling importance resampling particle filter tracking with multibag subspace restoration," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 264–276, Jan. 2018.
[42] Y. Wang, X. Ban, H. Wang, X. Li, Z. Wang, D. Wu, Y. Yang, and S. Liu, "Particle filter vehicles tracking by fusing multiple features," *IEEE Access*, vol. 7, pp. 133694–133706, 2019.
[43] J. Panda and P. K. Nanda, "Particle filter and entropy-based measure for tracking of video objects," in *Green Technology for Smart City and Society*, vol. 151, R. Sharma, M. Mishra, J. Nayak, B. Nayak, and D. Pelusi, Eds., Singapore: Springer, 2021, pp. 339–354.
[44] P. G. Bhat, B. N. Subudhi, T. Veerakumar, G. Di Caterina, and J. J. Soraghan, "Target tracking using a mean-shift occlusion aware particle filter," *IEEE Sensors J.*, vol. 21, no. 8, pp. 10112–10121, Apr. 2021.
[45] P. G. Bhat, B. N. Subudhi, T. Veerakumar, and S. Esakkirajan, "A fully automatic feature-based real-time traffic surveillance system using data association in the probabilistic framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11088–11097, Aug. 2022.
[46] M. Zolfaghari, H. Ghanei-Yakhdan, and M. Yazdi, "Real-time object tracking based on sparse representation and adaptive particle drawing," *Vis. Comput.*, vol. 38, no. 3, pp. 849–869, Mar. 2022.
[47] J. García, A. Gardel, I. Bravo, J. L. Lázaro, and M. Martínez, "Tracking people motion based on extended condensation algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 606–618, May 2013.
[48] M. Talha and R. Stolkin, "Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data," *IEEE Sensors J.*, vol. 14, no. 1, pp. 159–166, Jan. 2014.
[49] X. Lu, L. Song, S. Yu, and N. Ling, "Object contour tracking using multi-feature fusion based particle filter," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Singapore, Jul. 2012, pp. 237–242.
[50] F. Xu and L. Zhao, "A particle filter tracking algorithm based on adaptive feature fusion strategy," in *Proc. 10th World Congr. Intell. Control Autom.*, Beijing, China, Jul. 2012, pp. 4612–4616.
[51] D. Ding, Z. Jiang, and C. Liu, "Object tracking algorithm based on particle filter with color and texture feature," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 4031–4036.
[52] M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, "Motion history image: Its variants and applications," *Mach. Vis. Appl.*, vol. 23, no. 2, pp. 255–281, Mar. 2012.
[53] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Mach. Vis. Appl.*, vol. 13, no. 3, pp. 174–184, Jul. 2002.
[54] J. F. Aherne, A. N. Thacker, and I. P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.
[55] S. Dubuisson and C. Gonzales, "A survey of datasets for visual tracking," *Mach. Vis. Appl.*, vol. 27, no. 1, pp. 23–52, Jan. 2016.
[56] M. Firouznia, J. A. Koupaei, K. Faez, G. A. Trunfio, and H. Amindavar, "Adaptive chaotic sampling particle filter to handle occlusion and fast motion in visual object tracking," *Digit. Signal Process.*, vol. 134, Apr. 2023, Art. no. 103933.

JYOTIRANJAN PANDA (Member, IEEE) received the B.Tech. degree in electronics communication engineering and the M.Tech. degree in communication engineering, in 2012. He is currently pursuing the Ph.D. degree in video object tracking with Siksha 'O' Anusandhan, Deemed to be University, Bhubaneswar, Odisha, India. He is an Assistant Professor with the Department of CIOT, Siksha 'O' Anusandhan, Deemed to be University. His research interests include video object detection and tracking and computer vision.

PRADIPTA KUMAR NANDA (Senior Member, IEEE) received the degree (Hons.) in electrical engineering from the VSS University of Technology (UCE), Burla, Odisha, in 1984, the master's degree in electronics systems and communication engineering from the National Institute of Technology, Rourkela, Odisha, in 1989, and the Ph.D. degree in computer vision from IIT Bombay, in 1996. From 1986 to 2007, he was a Faculty Member with NIT Rourkela, India, with the last position as a Professor and the Head of the Department. He was instrumental in establishing the Center of Excellence on Industrial Electronics and Robotics and as the Principal Investigator. After serving NIT Rourkela, he joined Siksha 'O' Anusandhan, Deemed to be University, where he is currently discharging his duty as the Vice Chancellor. He was an Academic Visitor with the School of Computing, University of Leeds, U.K., in March 2006, and delivered lectures to their faculties and students. He has delivered several invited lectures in different international/national conferences, workshops/ summer, and winter schools. He has published 101 papers in various journals and conference proceedings. He has authored five research books and seven book chapters. His research interests include image processing and analysis, bio-medical image analysis, video-tracking, computer vision, soft computing and their applications, and ad-hoc wireless sensor networks. He is a fellow of IETE, India, a member of IET, U.K., and a Life Member of ISTE, India. He was the Chair of the IEEE GRSS Society, Kolkata Chapter.

TUHINANSU PRADHAN is currently pursuing the Ph.D. degree with the Kalinga Institute of Industrial Technology, Bhubaneswar, India. He is an Assistant Professor with the Institute of Technical Education and Research, Siksha 'O' Anusandhan, Deemed to be University. He is involved in teaching the Internet of Things using Python on Raspberry Pi to B.Tech. students.