Article

Adaptive Framework for Multi-Feature Hybrid Object Tracking

Ahmad Saeed Khattak , **Gulistan Raja and Nadeem Anjum**

Department of Electrical Engineering, University of Engineering and Technology, Taxila 47050, Pakistan; gulistan.raja@uettaxila.edu.pk (G.R.); nadeem.anjum@uettaxila.edu.pk (N.A.)

* Correspondence: ahmad.saeed.khattak@gmail.com; Tel.: +92-321-581-8008

Received: 22 October 2018; Accepted: 13 November 2018; Published: 19 November 2018

Abstract: Object tracking is a computer vision task deemed necessary for high-level intelligent decision-making algorithms. Researchers have merged different object tracking techniques and discovered a new class of hybrid algorithms that is based on embedding a meanshift (MS) optimization procedure into the particle filter (PF) (MSPF) to replace its inaccurate and expensive particle validation processes. The algorithm employs a combination of predetermined features, implicitly assuming that the background will not change. However, the assumption of fully specifying the background of the object may not often hold, especially in an uncontrolled environment. The first innovation of this research paper is the development of a dynamically adaptive multi-feature framework for MSPF (AMF-MSPF) in which features are ranked by a ranking module and the top features are selected on-the-fly. As a consequence, it improves local discrimination of the object from its immediate surroundings. It is also highly desirable to reduce the already complex framework of the MSPF to save resources to implement a feature ranking module. Thus, the second innovation of this research paper introduces a novel technique for the MS optimization method, which reduces its traditional complexity by an order of magnitude. The proposed AMF-MSPF framework is tested on different video datasets that exhibit challenging constraints. Experimental results have shown robustness, tracking accuracy and computational efficiency against these constraints. Comparison with existing methods has shown significant improvements in term of root mean square error (RMSE), false alarm rate (FAR), and F-SCORE.

Keywords: particle filter; mean shift; object tracking; multi-feature; Kalman filter

1. Introduction

The increase in the computational power of existing systems has led to huge investments in automated data analysis. Object tracking is one such class of algorithms that automatically locates the region of interest, possibly obscured by challenging constraints [1]. These constraints are what defines the requirements that should be considered while developing real-time robust object tracking algorithms. Earlier methods successfully tracked objects from stationary cameras using background subtraction techniques [2]. These methods, when combined with data association techniques, can track multiple objects [3]. However, in these methods, the scene structure is known in advance. Fukunage and Hostetlar introduced a method based on meanshift (MS) estimation using gradient descent that follows a reference template/model [4]. Later Comaniciu, Ramesh, and Meer used MS to solve the problem of tracking [5] and ever since it has shown continuous presence in the computer vision community [6–9].

MS is a non-parametric method that determines the location of the object by measuring distance between the histograms of the target and candidate templates. This is achieved by maximizing a similarity measure, the Bhattacharyya coefficient, until it converges. The MS method is robust

against partial occlusion and can adapt to size/scale variation under expensive mixture of Gaussian (MoG) [8,10,11]. Nevertheless, since histogram disassociates the target from the neighbourhood pixel information, it fails under the constraints such as fast object motion, clutter background, and full occlusion. This constrains the object from large displacements in consecutive frames, which as a consequence mandates some part of the object to lie in the basin of search. When the object is lost, the convergence of MS either becomes very slow or it eventually loses the object. MS also becomes slow when the size of the object increases.

On the other hand, the statistical category treats the tracking problem as a recursive computation in state space. Kalman filters (KF) and particle filters (PF) are the most popular methods in this category. KF give optimal estimation under the assumption of a linear state transition and Gaussian measurement and process noise [12]. PF, a popular statistical approach, waives off the restrictive hypothesis of KF and estimates the posterior density by combining random samples associated with weights [13–21]. Nummiaro et al. [16], Israd et al. [18] and Pe’rez et al. [19], are some of the most prominent research that has used color histograms with the PF method for object tracking. In [16], a method was developed that uses color with the PF method and the results are compared with the MS method and MS in combination with KF. The color-based PF method showed good performance against occlusion and non-linear object motion. The method of [18] used color and shape to represent the target. However, the method was successful in multi-object tracking, but the tracker distracts from the target when multiple people cross it. Most of these methods use global histogram for object representation and in the process the spatial information is lost. Pe’rez et al. introduced a technique that models multi-part of the target to compensate for some of the spatial information [19]. The method is robust against clutter background, size/scale variation and occlusion [19]. All these research work has proved robustness and accuracy of the PF method over MS and KF, however, the accuracy is dependent on a large number of particles. This dependency is overcome through embedding the MS optimization into the PF framework and that give rise to a new class of hybrid object tracking algorithms which will be discussed next.

2. Materials and Methods

This section discusses hybrid object tracking algorithm based on embedding MS into the PF methodology (MSPF). In this technique, the MS optimization procedure replaces the expensive and inaccurate particle validation process of the PF method. The MS optimization method reduces the particle count by finding the local modes for each particle. That essentially increases the accuracy of the particles’ state, which consequently reduces the need for a large number of particles.

There is a volume of research that approves the accuracy and robustness of MSPF based hybrid tracking methods [22–29]. Shan Caifeng combined PF and MS to track a hand for an application of intelligent wheelchair. Color and motion cues were used in a complementary fashion to offset the error of feature that is not discriminating the object [23]. The results of which are robust against clutter background and light illumination variation. In the same hybrid category, Anbang Yao introduced a particle filter based kernel object tracking (PFKBOT) algorithm. It is based on an incremental Bhattacharyya dissimilarity into the MS optimization method [24]. The idea was to continuously distinguish the background particles from the ones lying in the target region. And, consequently, the result shows robustness to background clutter and occlusion, however, at the cost of large computational power. Likewise, color is complemented with the motion information to solve the problem of background clutter and mild illumination change [26]. A Two Stage Hybrid Tracker (TSHT) was developed that uses color histograms in combination with orientation edge histograms to include the shape and inner edges of the object [27]. The results, however, fail under occlusion but nevertheless are robust towards changes in size/scale and fast motion. TSHT uses 5 MS iterations per particle to compute probable locations, therefore it does not meet the real-time requirements. Summary of the literature review is given in Table 1.

Table 1. Summary of state-of-the-art hybrid object tracking algorithms.

Ref.	Search Mechanism	Features Models	Robustness Towards Various Constraints
[5]	Deterministic (Mean shift)	Color + Texture	Partial occlusion, clutter, and size/scale
[6]		Color	Partial occlusion, size and scale
[7]		Edge + Texture	Small objects
[8]		Color	Orientation scale and position
[9]		Color	Fast object motion, partial occlusion
[10]		Color	Full occlusion, size and scale
[15]	Statistical (Particle Filters)	Color	Non-rigid deformations, partial occlusions and cluttered background
[16]		Color	Clutter background, Occlusion, size/scale variation and light illumination
[17]		Color and edge orientation	Clutter background and short time period occlusion
[18]		Color and shape	Large object motion, partial occlusion
[19]		Color	Cluttered background, occlusion and size/scale variation
[28]		Motion model	Illumination variation and partial occlusion
[20]	MSPF based hybrid systems	Color + motion Cue	Fast motion, light clutter, illumination change
[21]		Color	Multiple hypothesis, clutter background
[22]		HSV Color	Clutter background, size and scale
[23]		Color + Motion model	Occlusion, clutter and fast motion
[24]		Color + motion model	Fast object motion and clutter background
[25]		Color	Background clutter, full occlusion
[26]		HSV color components	Size/Scale, fast object motion, occlusion
[27]		Color + Edge Orientation histogram	Size/Scale, fast motion
[29]		Color + Motion model	Clutter background, light illumination variation and full occlusion
[30]		Color + local integral orientation	Scale and pose variation

These approaches rely on a combination of predetermined fixed features neglecting the adaptive nature of the background that might be caused by the moving object. This creates an impasse when the object moves toward a background that camouflage the object with either a similar texture or any other abrupt change. Consequently, the maneuvering aspects of the object constraints the algorithm. This proposed research work takes into account the above-mentioned shortcomings and presents the following two novelties:

1. The proposed AMF-MSPF framework implements a feature ranking module on top of the MSPF methodology. Thereby an adaptive multi-feature framework is implemented that selects the required features on-demand as and when required. Consequently, this enables the object tracking algorithm to discriminate the object locally. The feature ranking module re-initializes the MS procedure with new features and is triggered based on whether re-sampling occurs or not. When resampling occurs, a new set of N_f features are selected using the ranking module that are used for updating the target model.
2. As the PF algorithm itself is a very compute intensive method, embedding an MS into its particle validation process increases its computational load. Thereby it is pertinent that the complexity of the MS method is reduced to enable the proposed framework to run in real-time. We propose a novel MS optimization method based on an observation that MS only requires a fraction of sample to accurately track. This has led to huge reduction in computational load without inducing any significant error.

The rest of the paper is organized as follows: The mathematical formulation of the proposed AMF-MSPF framework is explained in Section 3. The experimental results are presented in Section 4. The concluding remarks along with future directions are summarized in Section 5.

3. Proposed Framework

The AMF-MSPF hybrid framework is developed in this section. In this framework, multi-features are adaptively selected from a large pool that is used by the MSPF method. Initially, a set of particles and their associated weights are generated using dynamic state equations (i.e., Equations (16) and (17)). A template of the object is initialized from these particles and processed through the ranking module to select the top N_f features. The MS optimization uses these features to validate the particles by herding them to more precise locations or state. The new state of the particles reinforces PF with more accurate measurements. A mathematical formulation, as shown in the Figure 1, is developed in this subsection.

3.1. Feature Ranking

Unlike the domains of image understanding and medical applications where offline feature ranking and selection are successfully adopted, the broad spectrum of applications in the computer vision domain is directly tied to real-time processing. We have used a simple ranking criterion adopted by Collins [9] to accommodate multiple features that are considered by the tracker on the fly. The ranking module ranks a pool of features based on the variance ratio between the background and foreground. The top ranked features are input into the AMF-MSPF framework, as a result it takes care of the changing variations such as background clutter and abrupt illumination changes. In this subsection, we describe the feature ranking module. Our feature space $\mathbb{F}$ is formed by a linear combination of the RGB color components having coefficients c_1 , c_2 , and c_3 :

$$\mathbb{F} = \{c_1R + c_2G + c_3B\} \quad (1)$$

where, $c \in [-2 \ -1 \ 0 \ 1 \ 2]$ and a combination of which produces 5^3 possible candidates in our feature space. Filtering out the redundant and useless cases, leave us with around 50 features. Let $fg(i)$ and

$bg(i)$ be the normalized discrete densities of background and foreground respectively. These densities are discretized to 16 bits for efficiency. The likelihood ratio of these densities is given by Equation (2).

$$\ell(i) = \log \frac{fg(j)}{bg(j)} \quad (2)$$

Figure 1. Adaptive multi-feature framework for meanshift based Particle Filter (AMF-MSPF).

Multiple likelihood images are generated using these likelihood ratios that are to be used by the MS procedure. We have used the traditional definition of variance, $\text{var}(x) = E(x)^2 - E(x)^2$, to calculate the variance of $\ell(i)$ w.r.t the object and background densities to maximize the inter-class variance between them. Equations (3) and (4) are used to calculate the variance:

$$v(\ell; fg) = E[\ell^2(j)] - (E[\ell(j)])^2 = \sum_j fg(j)\ell^2(j) - [\sum_j fg(j)\ell(j)]^2 \quad (3)$$

$$v(\ell; bg) = E[\ell^2(j)] - (E[\ell(j)])^2 = \sum_j bg(j)\ell^2(j) - [\sum_j bg(j)\ell(j)]^2 \quad (4)$$

In the final ranking step, Equation (5) finds the Variance Ratio (VR). Essentially, the inter-class variance of the foreground and the background is maximized for the feature space F .

$$VR(L; fg, bg) = \frac{v(\ell; (fg + bg)/2)}{[v(\ell; fg) + \text{var}(\ell; bg)]} \quad (5)$$

This variance ratio is sorted for the top N_f features which indicates the highest discrimination score. The likelihood, corresponding to features with the highest score, are used to form new likelihood images. These images are used to initialize the MS optimization procedure that produces new N_f locations for each particle. Symbolically Equations (1)–(5) is represented as : $N_f = \text{featureRnk}(F)$; the ranking module that produces N_f top features. The MS optimization procedure is explained in the next subsection.

3.2. MS Optimization Procedure

The MS procedure is a non-parametric gradient descent method used to move the particles' state, s_{k+1}^i to their new state $\hat{s}_{k+1}^i$ that is more precise and accurate. MS is initialized in each N_f features and applied to all the particles that essentially gives us as many new locations for each particle. These N_f locations are merged, over each particle, using median over them. Evaluating the MS procedure for every particle using multi-features is apparently an intensive computation. However, in the proposed MS optimization technique, the complexity is reduced by an order of magnitude. The weights are estimated for each pixel of the object that is first segmented using normalized cut algorithm based on J. Shi and J. Malik [31]. We pick only a fraction of samples from the reference and candidate regions according to these estimates.

Essentially, the proposed MS operates only on a fraction of samples. This improvement reduces its complexity by an order of magnitude. This subsection describes our proposed innovation introduced in the MS optimization method. The heart of any MS procedure is the maximization of the Bhattacharyya coefficient that is given by Equation (6):

$$\rho_{1:N_f} \equiv \rho[\mathbf{p}(y), \mathbf{q}] = \sum_{u=1}^m \sqrt{\mathbf{p}(y) \cdot \mathbf{q}_u} \quad (6)$$

The Bhattacharyya coefficient, $\rho_{1:N_f}$, is maximized over all N_f features. $\mathbf{q}_u$ and $\mathbf{p}_u$ are m -bin discrete color histograms of the target and candidates respectively. These bins are simply a series of small intervals that divides the whole range into smaller ones to reduce computational load. These densities are given from Equation (7) through Equation (8):

$$\mathbf{q}_u = C \sum_{i=1}^{N_h} k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (7)$$

$$\mathbf{p}_u(y) = C \sum_{i=1}^{N_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (8)$$

where, $\{x_i^*\}_{i=1 \dots N_h}$ and $\{x_i\}_{i=1 \dots N_h}$ are the pixels of the target and candidates respectively. δ is the delta function equal to 1 only at the particular bin u and 0 otherwise. N_h represents a fraction of random samples inside the target and candidate. In order to represent $\mathbf{q}_u$, $\mathbf{p}_u$ as densities, we multiply by a coefficient C to restrict the summations $\sum_{u=1}^m \mathbf{p}_u$ and $\sum_{u=1}^m \mathbf{q}_u$ to 1. k is a monotonic decreasing convex kernel. Traditionally Bhattacharyya coefficient is maximized using densities that are spatially weighted by an Epanechnikov kernel. However, these multivariate kernels are not good at dealing with non-rigid objects. We instead, employed a monotonic decreasing function to select a fraction of samples picked from the object and its immediate surroundings for spatial weighting. By using only a fraction of samples, the computational load reduces considerably. Equation (9) computes the

similarity measure based on the distance between the target and candidates using a fraction of samples from them.

$$d(y) = \sqrt{1 - \rho[\mathbf{p}(y), \mathbf{q}]}$$
 (9)

The MS procedure maximizes Equation (6), which consequently minimizes Equation (9). The MS optimization process is an iterative process and is initialized in the previous frame with the target position y_0 . The new location y_1 is evaluated based on the convergence of Equation (6). By expanding the Taylor series around coefficient $\mathbf{p}_u(y_0)$, the linear approximation of Equation (6) is obtained after some manipulation [6–8]:

$$\rho_{1:N_F}[\mathbf{p}(y), \mathbf{q}_u] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\mathbf{p}_u(y_0) \cdot \mathbf{q}_u} + \frac{1}{2} \sum_{u=1}^m \mathbf{p}_u(y) \sqrt{\frac{\mathbf{q}_u}{\mathbf{p}_u(y_0)}} \quad (10)$$

y substituting Equation (8) in Equation (10), we obtain

$$\rho_{1:N_F}[\mathbf{p}(y), \mathbf{q}_u] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\mathbf{p}_u(y_0) \cdot \mathbf{q}_u} + \frac{C}{2} \sum_{i=1}^{N_h} \omega_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (11)$$

where, $\omega_i = \sum_{u=1}^m \sqrt{\frac{\mathbf{q}_u}{\mathbf{p}_u(y_0)}} \delta[b(x_i) - u]$.

The first term, of Equation (11), is actually the previous location of the target which is not dependent on the new coordinate y . Therefore, we only maximize the second term of the equation to get the new target location y_1 .

$$y_1 = \frac{\sum_{i=1}^{N_h} x_i \omega_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{N_h} \omega_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \quad (12)$$

where $g(x)$ is a constant for the kernel profile that reduces Equation (12) to a simple weighted distance average as in Equation (13).

$$y_1 = \frac{\sum_{i=1}^{N_h} x_i \omega_i}{\sum_{i=1}^{N_h} \omega_i} \quad (13)$$

Update $\mathbf{p}(y_1)$ and evaluate $\rho_{1:N_F}[\mathbf{p}(y), \mathbf{q}]$ until $\|y_1 - y_0\| < \epsilon$ for every N_F feature and ϵ is usually 1 pixel. The particles along the gradient direction of the MS vector move to the local extreme of a probability density function and that determines the best location i.e., where Bhattacharyya coefficient is the highest. The MS procedure is applied to every particle over all the top N_F features given as in Equation (14).

$$[\mathbf{y}_{1:N_F}^{1:N_s}] = \text{MnSft}^{1:N_F}(\mathbf{s}_{k+1}^{1:N_s}) \quad (14)$$

$\mathbf{y}_{1:N_F}^{1:N_s}$ are the new N_F estimate locations, for each N_s particles, that are combined to form new estimates using Equation (15).

$$[\hat{\mathbf{s}}_{k+1}^{1:N_s}] = \mu\text{edian}(\mathbf{y}_{1:N_F}^{1:N_s}) \quad (15)$$

$\hat{\mathbf{s}}_{k+1}$ are the new estimate of the particles that are now closer to their local maxima than $\mathbf{s}_{k+1}$. Figure 1 depicts the working of our framework. The top N_F features along with the initialized particles serves as the input to the MS modules. This gives us new probable locations of the object for all the selected features. Or in other words, the particles after processing through MS, gives rise to N_F locations for each particle. The median over all the locations gives us the particles that are more accurate i.e., $\hat{\mathbf{s}}_{k+1}$. These new hypotheses are used to estimate the posterior density through Equation (20) in the next subsection.

3.3. MS Embedded Particle Filter

The MSPF hybrid method is based on the Bayesian framework which basically deals with the evolution of object state. The belief network is reinforced by the measurement process through a set of dynamic equations as follows:

$$\mathbf{s}_{k+1} = \mathbf{f}_k(\mathbf{s}_k + \gamma_k) \quad (16)$$

$$\mathbf{z}_{k+1} = \mathbf{h}_k(\mathbf{s}_{k+1} + \eta_k) \quad (17)$$

where, $\mathbf{s}_k$ and $\mathbf{s}_{k+1}$ are the object states at time k and $k + 1$. $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$, are the dynamic equations for drawing particles and taking new measurements $\mathbf{z}_{k+1}$. γ_k and η_k represent the process and measurement noise respectively. The object state distribution is estimated based on all the previous measurements that theoretically are:

$$p(\mathbf{s}_{k+1}|\mathbf{z}_{1:k}) = \int p(\mathbf{s}_{k+1}|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{z}_{1:k})d\mathbf{s}_k \quad (18)$$

This prior is used in the prediction step when the measurement $\mathbf{z}_{k+1}$ is available. Bayes' rule recursively updates this prior in the prediction step as in Equation (19):

$$p(\mathbf{s}_{k+1}|\mathbf{z}_{1:k+1}) = \frac{p(\mathbf{z}_{k+1}|\mathbf{s}_{k+1})p(\mathbf{s}_{k+1}|\mathbf{z}_{1:k})}{\int p(\mathbf{s}_{k+1}|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{z}_{1:k})d\mathbf{s}_k} \quad (19)$$

where, $p(\mathbf{z}_{k+1}|\mathbf{s}_{k+1})$ is the likelihood distribution and $\int p(\mathbf{s}_{k+1}|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{z}_{1:k})d\mathbf{s}_k$ is the normalization factor. The posterior density function is approximated through summations or weighted summation over all particles, as in Equation (20):

$$p(\mathbf{s}_{k+1}|\mathbf{z}_{1:k+1}) \sim \sum_{i=1}^{N_s} \mathbf{w}_{k+1}^i \left(\mathbf{s}_{k+1}^i - \hat{\mathbf{s}}_{k+1}^i \right) \quad (20)$$

where, $\hat{\mathbf{s}}_{k+1}^i$ are the new particles obtained by processing $\mathbf{s}_{k+1}^i$ through the MS procedure using Equations (14) and (15). So now, effectively the PF particle validation process is replaced with the MS procedure by inserting $\hat{\mathbf{s}}_{k+1}^i$ into the posterior distribution function in Equation (20). The vector $\mathbf{w}_{k+1}^i$ are the weights associated with each particle and are calculated through Equation (21).

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i \frac{p(\mathbf{z}_{k+1}|\mathbf{s}_{k+1}^i)p(\mathbf{s}_{k+1}^i|\mathbf{s}_k^i)}{q(\mathbf{s}_{k+1}^i|\mathbf{s}_k^i, \mathbf{z}_k)} \quad (21)$$

Usually the likelihood distribution is used to calculate the weights of the particles:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i \left(p(\mathbf{z}_{k+1}|\mathbf{s}_{k+1}^i) \right) \quad (22)$$

In order to get normalized weights, $\mathbf{w}_{k+1}^i$ is divided by sum of all weights:

$$\mathbf{w}_{k+1}^i = \frac{\mathbf{w}_{k+1}^i}{\sum_{i=0}^{N_s} \text{sum}(\mathbf{w}_{k+1}^i)} \quad (23)$$

Expectation is used to approximate the posterior densities of Equation (20) given as:

$$\mathbb{E}(\mathbf{s}_{k+1}|\mathbf{z}_{1:k+1}) \sim \frac{1}{N_s} \sum_{i=1}^{N_s} \left(\mathbf{w}_{k+1}^i \mathbf{s}_{k+1}^i \right) \quad (24)$$

We should be very careful about the degeneration phenomenon, also called sample impoverishment, in which a few particles assume the leading role in approximating the posterior. This leads to a situation where many particles will cluster together as a result of which the sample

set will have repeated particles. This results in a large percentage of particles getting insignificant tiny weights. Consequently, the tracker loses the target due to particles drifting towards one side. We calculate $\mathcal{N}_{eff}$, a reasonable measure of degeneracy, using Equation (25)

$$\mathcal{N}_{eff} = \frac{1}{\sum_{i=0}^{N_s} \mathbf{w}_{k+1}^i} \quad (25)$$

So, IF $\mathcal{N}_{eff} \leq N_T$ we redistribute the clusters of particles, so that new particles are generated out of this cluster and re-initialize their weights to $\frac{1}{N_s}$. N_T is a threshold that triggers re-sampling if it is greater than $\mathcal{N}_{eff}$. The idea behind resampling is to get rid of particles with tiny weights and generate more particles from the ones with greater weights. As resampling can only execute after the weight calculation and normalization is done, it becomes a bottleneck in parallelizing the PF computations. Re-sampling although reduces the effects of degeneration, however, it introduces loss of diversity among the particles, also called impoverishment due to the fact that the resultant particles are clustered to close vicinity [21,32]. This is because redundant particles are chosen from the same points and hence the tracker has a chance to lose the object. To mitigate from this problem, the proposed framework executes the feature ranking module to update the model of the object alongside re-sampling. The next subsection presents the pseudo code of the proposed AMF-MSPF framework.

3.4. Pseudo Code of AMF-MSPF

At time k, execute the following steps:

1. *Particle initialization step:*
 $\{i = 1 : N_s; k = 1; \text{Flag} = \text{True}\}$
 $\{(\mathbf{s}_k^i, \mathbf{w}_k^i)\}_{i=1}^{N_s}$ to give $[\{\mathbf{s}_1^{1:N_s}, \mathbf{w}_1^{1:N_s} = (\frac{1}{N_s})_1^{1:N_s}\}]$
2. *Feature Ranking step:*
 $[N_F] = \text{featureRnk}(\mathbb{F})$
3. *Propagation step:*
 $\{k = 2, 3, 4, \dots\}; p(\mathbf{s}_{k+1} | \mathbf{s}_k) = (\mathbf{s}_{k+1}^{1:N_s}, \mathbf{w}_{k+1}^{1:N_s})\}$
4. *MS Optimization Step:*
 $[\mathbf{y}_{1:N_F}^{1:N_s}] = \text{MnSft}^{1:N_F}(\mathbf{s}_{k+1}^{1:N_s})$
 $[\hat{\mathbf{s}}_{k+1}^{1:N_s}] = \text{median}(\mathbf{y}_{1:N_F}^{1:N_s})$
5. *Weight Calculation and normalization step:*
 $\mathbf{w}_{k+1}^{1:N_s} = \mathbf{w}_k^{1:N_s} * p(\mathbf{z}_{k+1} | \mathbf{s}_{k+1}^{1:N_s})$
 $\mathbf{w}_{k+1}^{1:N_s} = \frac{\mathbf{w}_{k+1}^{1:N_s}}{\sum_{i=0}^{N_s} \mathbf{w}_{k+1}^i}$
 $\mathcal{N}_{eff} = \frac{1}{\sum_{i=0}^{N_s} \mathbf{w}_{k+1}^i * \mathbf{w}_{k+1}^{it}}$
6. *Estimation Step: (Posterior Estimation)*
 $p(\mathbf{s}_{k+1} | \mathbf{z}_{1:k+1}) \sim \sum_{i=1}^{N_s} \mathbf{w}_{k+1}^i (\mathbf{s}_{k+1}^i - \hat{\mathbf{s}}_{k+1}^i)$
7. *Re-sampling Step: (Particle redistribution and weight re-initialization)*
 IF $\mathcal{N}_{eff} \leq N_T$
 $\{\mathbf{s}_{k+1}^{1:N_s}, \mathbf{w}_{k+1}^{1:N_s}\} = \{\mathbf{s}_{k+1}^{1:N_s}, (\frac{1}{N_s})_1^{1:N_s}\}$
 GOTO STEP 2 and REPEAT
 ELSE
 GOTO STEP 3 and REPEAT

4. Experimental Results

This section evaluates and compares the proposed AMF-MSPF framework with the TSHT of [27] and PFKBOT of [24]. The choice of selecting these reference approaches, is because they employ multi-features and an adaptive model updating techniques similar, in some way, to our multi-feature framework. The experiments have been conducted on frames of different sizes and processing is done using an Intel Core i5 2.60 GHZ with 4 GB RAM. To illustrate the efficiency of the proposed AMF-MSPF, we applied it to video sequences having full occlusions, abrupt intensity changes, and clutter background introduced by the moving object. These challenging sequences evidently makes the experiments very difficult. The accuracy and robustness of the proposed AMF-MSPF are tested on the WalkByShop1cor, CAVIAR, and PETS video data sets. Table 2 highlights some of the important characteristics of these sequences.

Table 2. Description of video sequence.

Video Sequence	Description	Characteristic	Frame Size	No. of Frames
WalkByShop1cor	Couple walking along a corridor browsing	Regular mild and Severe occlusion	384×288	2359
Browse4	Person moves in an area with abrupt intensity change	abrupt illumination change with non-linear motion	384×288	1138
PETS 2007	People walking in crowd with different obstacles	Severe occlusion by abrupt intensity variations	576×720	3000

The reason for selecting these data sets is because they have rich characteristics such as full occlusion, clutter, and abrupt illumination change. The computer vision research community widely uses these data sets to evaluate their algorithms. The prime goal, of the simulations, is to track manually initialized regions of interest during long video sequences under clutter background and occlusions by intensity. The ground truth locations of the objects are recorded manually for each sequence. Red, blue, green and yellow depicts the results of the ground truth, proposed AMF-MSPF, PFKBOT, and TSHT respectively. For a quantitative analysis F-SCORE, false alarm rate (FAR) and root mean square error (RMSE) are evaluated for all the sequences under test, as shown in Table 3.

Precision (P) and recall (R) are given by Equations (26) and (27):

$$P = \frac{TP}{TP + FP} \quad (26)$$

$$R = \frac{TP}{TP + FN} \quad (27)$$

where, TP, FP, FN and TN are the number of true positives, false positives, false negatives, and true negatives respectively. F-SCORE becomes trivial if we know P and R that is given by Equation (28):

$$F_SCORE = 2 * \frac{P * R}{P + R} \quad (28)$$

Similarly, FAR is calculated as the ratio between the number of negative event wrongly categorized as positive and the total number of actual negative events using Equation (29).

$$FAR = \frac{FP}{TN + FP} \quad (29)$$

4.1. Visual Tracking Result

We start with the WalkByShop1cor dataset which is a long video sequence exhibiting several constraints. The dataset exhibits multiple full occlusions in addition to similar object appearance in

frames 350–410 and 870–930. The results of AMF-MSPF are close to the ground truth and outperforms TSHT and PFKBOT methods as shown in Figure 2. PFKBOT fails on encountering instances having a background similar to the object. The error accumulates until the PFKBOT loses the target because FP strays the particles to false mode. However, in THST, the object recovers from the occasional distraction by the background clutter. This is due to the fact that TSHT method takes into account both the spatiotemporal aspects of the object that leads the particles towards more likely modes. Under mild but regular intensity change, TSHT is quite comparable to the proposed AMF-MSPF in the WalkByShop1cor dataset. Figure 2 highlights the visual tracking results.

Figure 2. WalkByShop1cor sequence: mild but regular occlusion.

Since the results of AMF-MSPF and the reference methods have come in proximity, RMSE metric is evaluated for a closer look. In Table 3, the RMSE, of AMF-MSPF, THST and PFKBOT for WalkByShop1cor dataset are 5.76, 7.76 and 30.70 respectively. Consequently, the FAR of the proposed method is the lowest, due to very few false positives i.e., 0.13 as compared to 0.24 and 0.35 for THST and PFKBOT respectively. Moreover, the F-Score of the proposed method is 0.93 as compared to 0.86, 0.79 for THST and PFKBOT.

Table 3. RMSE, Accuracy and F-SCORE for all video sequences.

Video Sequence	RMSE			FAR			F_SCORE		
	PFKBOT	TSHT	Proposed	PFKBOT	TSHT	Proposed	PFKBOT	TSHT	Proposed
Browse4	55.80	41.30	9.18	0.49	0.47	0.17	0.67	0.71	0.91
WalkBy-Shop1cor	30.70	7.76	5.76	0.35	0.24	0.13	0.79	0.86	0.93
PETS-2007	24.70	15.30	11.12	0.35	0.27	0.15	0.71	0.79	0.91

In the next experiments, we consider two more sequences from the Browse4.mpg dataset. In these sequences, the object moves in severe intensity occluded areas. The abrupt intensity change occurs

in multiple instances from frames 230–410, frames 880–920 and frames 1005–1050. The experimental results show that under abrupt intensity change, the AMF-MSPF is robust and tracks the object, as highlighted in Figures 3 and 4. In Table 3, the RMSE of AMF-MSPF, TSHT, and PFKBOT is 9.18, 41.30 and 55.80 respectively. Low RMSE indicates FAR and consequently a higher F-SCORE that is 0.91 for the proposed.

Figure 3. Browse4 sequence: Occlusion by high intensity.

Figure 4. Browse4 sequence: Occlusion by high intensity.

In the final experiment, AMF-MSPF is tested on the PETS 2007 dataset, which exhibits continuous background change due to occlusion by intensity as well as other objects. The results of AMF-MSPF are convincing as compared to the TSHT and PFKBOT methods as shown in Figure 5. Contrasting TSHT and PFKBOT, the proposed AMF-MSPF framework selects a new set of N_F features that are used to initialize MS procedure for every particle. These features are selected by the ranking module that is triggered whenever the re-sampling step occurs due to the degeneration problem. In contrast to other methods, our method is more accurate because it re-initializes the object based on local features. For a quantitative analysis and highlights of some of the prominent characteristics of our method Tables 3 and 4 are given in this regard.

Figure 5. PETS 2007 sequence: Crowd.

Table 4. Results of various trackers under challenging constraint.

Constraint/Methods	[24]	[27]	Proposed Method
Full Occlusion	✓	✓	✓
Clutter Background	✗	✓	✓
Intensity Change	✗	✗	✓

4.2. Computational Complexity

After evaluating the proposed AMF-MSPF on several challenging video datasets, we obtained improved results in term of robustness and accuracy. However, the MSPF hybrid methodology implicitly has an additional cost of incorporating the MS optimization into the already complex PF algorithm. One innovation of the proposed research is the simplification of the MS optimization.

The simplification comes through an observation, that only a fraction of samples was required by the MS optimization procedure. The computational efficiency of the proposed MS was evaluated on a sequence extracted from the Browse4 dataset. As can be seen in Figure 6, the error introduced, due to dropping more than 75% of the samples, is negligible. And our simplified MS is able to track the object successfully in real-time, even under a Matlab implementation. This has led to a huge computational reduction in the MS method without significantly compromising its accuracy. This as a consequence saves resources for feature ranking module. Thereby, the proposed AMF-MSPF framework implements the feature ranking on top of the MSPF methodology without aggravating its computational complexity. The complexity of proposed framework is considerably reduced as compared to TSHT and PFKBOT because in those implementations, MS takes into account all the samples of object window. The AMF-MSPF framework can process 10–15 frames per second, which is perceived by humans, while none of the reference methods are able to run in real-time.

Figure 6. A sequence from the Browse4 dataset: (top to bottom, left to right) (a) Centre of the tracked object using full range vs. fraction of pixels; (b) Visual tracked sequence; (c) Score obtained taking into account full range vs. a fraction of pixels; (d) Error for selecting only a fraction of pixels.

Let us derive the overall computational cost of the proposed AMF-MSPF framework. Let I_{ter} be the total number of iteration for each MS procedure (in our case $I_{ter} = 5$), $T_{p_gen_wt}$ be the execution time required for sample generation and their associated weights, $T_{re-sampling}$, T_{ms} , and T_{rnk} be the execution times for the resampling, MS procedure and ranking module respectively. Then the execution time of the AMF-MSPF framework is given in Equation (29):

$$T_{Amf-mspf} = [T_{p_gen_wt} + T_{ms} + T_{re-sampling} + T_{rnk}] \quad (30)$$

Although MS is initialized in multiple features over all particles, however down sampling the number of pixels reduces its execution time by an order of magnitude. Thus with the reduced pixels, N_h , the complexity of our proposed MS is $O(m * N_f * N_h)$ which has no significant impact on Equation (29). This saves enormous computational power that is utilized by the ranking module. Since the MS optimization is applied multiple times to every particle until convergence, it moves the particles to new locations where it no longer conforms to the posterior distribution. This is what is known as the degeneracy problem in which a small number of particles dominate the weight race and therefore the estimates are heavily influenced by them. Consequently, the re-sampling stage is introduced to mitigate from the degeneracy problem by replicating particles with larger weights and removing the particles with negligible ones. This, however, constraints the system that re-sampling step be carried out after the particle weighting and normalization steps and that makes it a bottleneck in parallelizing the PF computations. The complexity of the re-sampling module is also equal to the particle generation and weighting step i.e., $O(N_s)$.

While the ranking module is quadratic in number of samples from the background and foreground i.e., $O(F * fg(j) * bg(j))$, nevertheless, it is only triggered when the re-sampling step is required. Since the re-sampling step is not carried in every step, most of the time the tracking is performed using only the novel optimized MSPF. Therefore, asymptotically speaking, this makes the overall complexity of our proposed AMF-MSPF framework equal to $O(n)$ with high probability.

5. Summary

In summary, this research work brings improvement to the field of robust object tracking in two novel ways. Firstly, it develops an adaptive multi-feature framework on top of the MSPF methodology. A ranking module ranks the given feature space based on maximizing the inter-class variance between the background and foreground. High variance score signifies the chances for better target discrimination and thereby the top features are selected for updating the target model. The likelihoods corresponding to these features are continuously used to form new likelihood images that dynamically initialize the MS to enable tracking in the local context. Secondly, the computational cost of the MS optimization method of the proposed AMF-MSPF framework is reduced. Consequently, that enables tracking in real-time unlike other methods in the hybrid tracking category that can only process a few frames.

The accuracy and robustness of the proposed AMF-MSPF is tested on the WalkByShop1cor, CAVIAR and PETS video data sets. These sequences are known for their challenging constraints that makes the experiments very difficult. The AMF-MSPF framework was found to provide improved tracking performance compared with other conventional methods based on hybrid object tracking methodology. The experimental results demonstrate successful visual tracking even under extreme intensity variations and full occlusion. The average F-SCORE (over all the video sets) of the AMF-MSPF is 0.92 as compared to 0.79 and 0.72 for TSHT and PFKBOT respectively.

The proposed AMF-MSPF implements a ranking module on top of the MSPF methodology to give it an additional layer of capability. Consequently, this has increased the performance of the MSPF that now takes into account any rapid movement of the object that changes its background. However, the improvement is tied to an assumption that the target reference model deviates significantly from its instance of initialization which is updated at the sampling step. This assumption needs to be carefully studied and there is a need to research on a more robust dynamic technique that switches the required features for the target update. In future research, we also plan to employ statistical estimation or deterministic background weighting mechanisms to boost the target in a cluttered background. As the clutter background eclipses some part of the target, and that can hamper the performance of the feature evaluation process. We have employed a linear combination of the RGB color components. However, the availability of a potentially large spatio-temporal feature space can also be experimented with to further improve tracking using the AMF-MSPF framework.

Author Contributions: A.S.K. conceived and implemented the idea, laid the mathematical formulation and wrote the original draft paper; G.R. supervised the overall project and reviewed the paper; N.A. analyzed the data and results, and verified the mathematical formulation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khattak, A.S.; Raja, G.; Anjum, N.; Qasim, M. Integration of Meanshift and Particle Filter: A Survey. In Proceedings of the 2014 12th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2014; pp. 286–291.
2. Sahoo, P.K.; Kanungo, P.; Parvathi, K. Three frame based adaptive background subtraction. In Proceedings of the 2014 International Conference on High Performance Computing and Applications (ICHPCA), Bhubaneswar, India, 22–24 December 2014; pp. 1–5.
3. Rasmussen, C.; Hager, G.D. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 560–576. [[CrossRef](#)]
4. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40. [[CrossRef](#)]
5. Comaniciu, D.; Ramesh, V.; Meer, P. Real-time tracking of non-rigid objects using mean shift. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000), Hilton Head Island, SC, USA, 15 June 2000; Volume 2, pp. 142–149.
6. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [[CrossRef](#)]
7. Yilmaz, A.; Shafique, K.; Shah, M. Target tracking in airborne forward looking infrared imagery. *Image Vis. Comput.* **2003**, *21*, 623–635. [[CrossRef](#)]
8. Yilmaz, A. Kernel-based object tracking using asymmetric kernels with adaptive scale and orientation selection. *Mach. Vis. Appl.* **2011**, *22*, 255–268. [[CrossRef](#)]
9. Collins, R.T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1631–1643. [[CrossRef](#)] [[PubMed](#)]
10. Fang, J.; Yang, J.; Liu, H.; Lv, J.; Zhou, Y. Robust fragments-based tracking with adaptive feature selection. *Opt. Eng.* **2010**, *49*. [[CrossRef](#)]
11. Dulai, A.; Stathaki, T. Mean shift through scale and occlusion. *IET Signal Process.* **2012**, *6*, 534–540. [[CrossRef](#)]
12. Gordon, N.; Ristic, B.; Arulampalam, S. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*; Artech House: London, UK, 2004.
13. Doucet, A.; Simon, G.; Christophe, A. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **2000**, *10*, 197–208. [[CrossRef](#)]
14. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proc. F-Radar Signal Process.* **1993**, *140*, 107–113. [[CrossRef](#)]
15. Czyz, J.; Ristic, B.; Macq, B. A particle filter for joint detection and tracking of color objects. *Image Vis. Comput.* **2007**, *25*, 1271–1281. [[CrossRef](#)]
16. Nummiaro, K.; Koller-Meier, E.; Van Gool, L. An adaptive color-based particle filter. *Image Vis. Comput.* **2003**, *21*, 99–110. [[CrossRef](#)]
17. Yang, C.; Duraiswami, R.; Davis, L. Fast multiple object tracking via a hierarchical particle filter. In Proceedings of the IEEE 2005 Computer Vision Conference, Beijing, China, 17–21 October 2005; Volume 1, pp. 212–219.
18. Isard, M.; MacCormick, J. BraMBLe: A Bayesian multiple-blob tracker. In Proceedings of the 2001 Computer Vision European Conference, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 34–41.
19. Patrick, P.; Hue, C.; Vermaak, J.; Gangnet, M. Color-based probabilistic tracking. In *Computer Vision on European Conference*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 661–675.
20. Dou, J.; Li, J. Robust visual tracking based on interactive multiple model particle filter by integrating multiple cues. *Neurocomputing* **2014**, *135*, 118–129. [[CrossRef](#)]
21. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]

22. Cheng, C.; Ansari, R. Kernel particle filter for visual tracking. *IEEE Signal Process. Lett.* **2005**, *12*, 242–245. [[CrossRef](#)]
23. Shan, C.; Tan, T.; Wei, Y. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognit.* **2007**, *40*, 1958–1970. [[CrossRef](#)]
24. Yao, A.; Lin, X.; Wang, G.; Yu, S. A compact association of particle filtering and kernel based object tracking. *Pattern Recognit.* **2012**, *45*, 2584–2597. [[CrossRef](#)]
25. Chia, Y.S.; Kow, W.Y.; Khong, W.L.; Kiring, A.; Teo, K.T.K. Kernel-based object tracking via particle filter and mean shift algorithm. In Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS), Melacca, Malaysia, 5–8 December 2011; pp. 522–527.
26. Wang, Z.; Yang, X.; Xu, Y.; Yu, S. CamShift guided particle filter for visual tracking. *Pattern Recognit. Lett.* **2009**, *30*, 407–413. [[CrossRef](#)]
27. Maggio, E.; Cavallaro, A. Hybrid particle filter and mean shift tracker with adaptive transition model. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, 23 March 2005; Volume 2, pp. 221–224.
28. Guo, W.; Zhao, Q.; Gu, D. Visual tracking using an insect vision embedded particle filter. *Math. Probl. Eng.* **2015**, *2015*, 573131. [[CrossRef](#)]
29. Yin, M.; Zhang, J.; Sun, H.; Gu, W. Multi-cue-based CamShift guided particle filter tracking. *Expert Syst. Appl.* **2011**, *38*, 6313–6318. [[CrossRef](#)]
30. Wei, Q.; Xiong, Z.; Li, C.; Ouyang, Y.; Sheng, H. A robust approach for multiple vehicles tracking using layered particle filter. *AEU-Int. J. Electron. Commun.* **2011**, *65*, 609–618. [[CrossRef](#)]
31. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
32. Li, T.; Sun, S.; Sattar, T.P.; Corchado, J.M. Figureht sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Syst. Appl.* **2014**, *41*, 3944–3954. [[CrossRef](#)]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).