

Self scale estimation of the tracking window merged with adaptive particle filter tracker

Azdoud Youssef, Amine Aouatif, Nassih Bouchra, Ngadi Mohammed

Systems Engineering Laboratory National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Nov 26, 2021

Revised Aug 4, 2022

Accepted Aug 30, 2022

Keywords:

Kernel distribution

Object tracking

Particle filter

Scale change

ABSTRACT

Tracking a mobile object is one of the important topics in pattern recognition, but it has some obstacles. A reliable tracking system must adjust their tracking windows in real time according to appearance changes of the tracked object. Furthermore, it has to deal with many challenges when one or multiple objects need to be tracked, for instance when the target is partially or fully occluded, background clutter, or even some target region is blurred. In this paper, we will present a novel approach for a single object tracking that combines particle filter algorithm and kernel distribution that update its tracking window according to object scale changes, whose name is multi-scale adaptive particle filter tracker. We will demonstrate that the use of particle filter combined with kernel distribution inside the resampling process will provide more accurate object localization within a research area. Furthermore, its average error for target localization was significantly lower than 21.37 pixels as the mean value. We have conducted several experiments on real video sequences and compared acquired results to other existing state of the art trackers to demonstrate the effectiveness of the multi-scale adaptive particle filter tracker.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Azdoud Youssef

Systems Engineering Laboratory, National School of Applied Sciences, Ibn Tofail University

Kenitra, Morocco

Email: youssef.azdoud@uit.ac.ma

1. INTRODUCTION

Tracking mobile object is a challenging and one of the important topics in the computer vision. Various applications were lead in this task, such as traffic monitoring [1], video surveillance [2], auto driving, braking assistance and human behavioral analysis [3]. Despite the existing research in the past decades, building a robust and effective tracking system continues to be a wide area of research in this digital area. Actually, what makes this research field a challenge is its problematic heterogeneity. Accordingly, every object tracking approach has to prove its performance in front of many constraints. In the literature, a large number of methods were using the same descriptors to represent objects, for instance color, shape, texture, some features are presented in a frequency space [4] in some case the features can be combined. In the literature, the tracking algorithm can be categorized into two classes, particularly deterministic approaches and probabilistic approaches. Deterministic approaches follow targets cross frames by searching iteratively for area most similar to the target window area via maximizing measure between those areas. A typical deterministic method is mean shift [5], it proposes a face tracking algorithm with an improved implementation of mean shift and identification of the face was based on Viola and Jones algorithm, they have also used corrected background weighted histogram to reduce noise on face on the process. More deterministic approaches were adopted by

integrating other feature description such as texture feature [6], scale-invariant feature transform (SIFT) [7], cross-bin metric [8], tracking by detection based on Hungarian algorithm [9], discriminative correlation filters [10]–[12], online multiple instance learning [13] and positive and negative learning bootstrapping binary classifiers [14]. In contrast probabilistic approaches use uncertainty and randomized logic to find the nearest solution. In this work, we adopted tracker algorithm based on a probabilistic approach, as they are the frequently used ones such as genetic algorithm (GA), cuckoo search (CS) algorithm, particle swarm optimization (PSO), and particle filter (PF). In this context, Maria *et al.* [15] introduce GA in hyperspectral images analysis, where they introduce a GA based on mutual information and normalized mutual information as fitness functions based on mutual information to achieve this bands selection. While Gao *et al.* [16] recommend CS algorithm to solve tracking problem, the algorithm has two parameters the number of nests and the probability discovering, it initializes the nest then it replaces them by Levy Flight random model while calculating the fitness of each nest. The best observation represents the target for the current frame. Furthermore, the PSO was also employed in visual tracking as stochastic optimization [17], [18], it is based on random particles initialization. These particles have two parameters, first the velocity that controls the movement, second the weight value calculated via fitness function. Regarding scale changing many methods have adopted mean-sift as base feature appearance. The algorithm proposed in [19] where Yu *et al.* have adopted mean shift to propose an adaptive hybrid non-rigid target tracking method that uses color histogram to process the characteristics. Yuan *et al.* [20] suggest correlation filter framework base on particle filter detection model, which reduce target loss by the tracker. Histogram of oriented gradients (HOG) feature [21] were used with correlation filter to learn an adaptive multi-scale. Furthermore, correlation filters framework were combined with deep learning approaches such as convolutional neural networks (CNN) [22], [23], new tracker category utilize Yolo and deep sort (DS) combination to detect and track the target [24], [25]. Circulant structure of tracking by detection with kernels (CSTDK) tracker proposed in [26] provides sufficient performance and good results in speed processing. The CSTDK was enhanced by applying the color features in [27]. However, CSTDK tracker proves its efficiency [28] by using HOG feature vector. At [29] multi-scale information measure calculates the object size change that update the scale ratio and integrated particle filtering tracking framework.

As we said previously probabilistic methods view the tracking algorithm as a state solving problem under the Bayesian framework, modelling uncertainty and propagating the conditional densities through the tracking process, in our case we decided to use particle filter as an optimisation algorithm with a strategic model to solve tracking problem by enhancing propagation process and handling multiple spatial scales and aspect ratios. The contributions of this work are summarized as: i) We propose a method that integrates kernel distribution at resampling phase to generate particles that could give accurate location during the tracking process, and ii) Increase the algorithm robustness, which is new, by combining scale updating process with particle filter to deal with target scale change on consecutive frames.

The remainder of paper is organized as: section 2.1 outlines a brief introduction of PF algorithm. Section we define image information measure, then we describe object tracking methods with self-updating window that use object information measure 2.2. Section 2.3 discusses our proposed method named multi-scale adaptive particle filter tracker (MSAPF) using kernel distribution. Experimental results are presented in section 3. Finally, section 4 depicts a conclusion.

2. METHOD

2.1. Particle filter

Particle filter or sequential Monte Carlo method's [30] estimates the unknown state x_t at time t from a group of observations $o_t^{1 \dots N}$ influenced by noise, which are presented as (1) and (2):

$$x_t = p(x_t | x_{t-1}) \quad (1)$$

$$o_t = h_t(x_t, v_t) \quad (2)$$

where x_t is the system state and o_t is the observation, v_t is the observation noise, h_t is the observation model, $p(x_t | x_{t-1})$ is the probability distribution of state x at t and $p(o_t | x_t)$ is the probability distribution of the observation o at t . The main aim of PF is finding a good approximation of the system state model which is

done by a set of weighted samples $X = \{(x^{(i)}, w^{(i)})\} : i = 1, \dots, N$. Where X is the set of samples. In order to find the best set of particles we use a proposal distribution $g(\cdot)$, according to the sampling principle, we calculate the differences between these distributions by assigning a proportional weight for each particle with the use of the function (3):

$$w_t^i = \frac{f(x_t^i)}{g(x_t^i)}, \quad (3)$$

with $f(\cdot)$ is the system distribution and $g(\cdot)$ is the proposal distribution. The weights values are reflecting the estimate of the probability distribution of the state at time t . In $t + 1$ the particle set is propagated by a dynamic model and weighted with a likelihood function w^i . Next, in the resampling step new unweighted particle set will be generated. This process is iterated each time until it finds the best estimate. In this work, we have chosen particle filter (PF) with a model tracking strategy for many reasons. Firstly, PF work well with high dimensional problems. Secondly, they are not computationally expensive. And, the algorithm is flexible to implement. Finally, it was effectively integrable with scale variation approaches.

2.2. Object's information measurement

The human vision perception exhibits a scale effect, the farther one sees an image, the less details he observes. Furthermore, in object tracking, we have more information in large scale than in small scale. Qian *et al.* [29] have proposed a method being able to measure the quantity of information of object, that was used to update the tracking window size, whenever the object information changes.

2.2.1. Image feature points types

In Qian *et al.* [29] work, they consider primal sketch, which is the first step of visual processing [31] as the main key point to measure image information. Since the primal sketch it represents a reflection of the physical reality, and it gives the number of elements on the image. Based on that two versions of image feature points were defined and were used to compute the multi-scale information. Considering that $f(x, y)$ is an image and $P(x, y)$ is a pixel on the image, and it has eight nearest neighbors $N_l(P)$ in eight l directions $l \in L, L = \{k\pi/4, k = -4, \dots, 3 : k \in Z\}$, as shown in Figure 1.

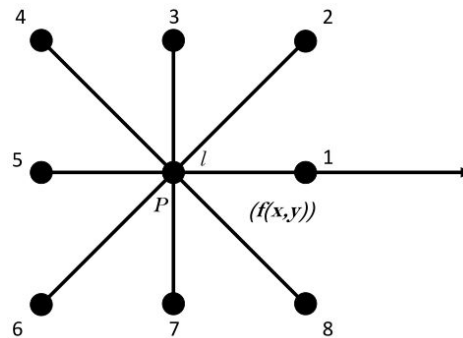


Figure 1. The P pixel and its eight nearest neighbors

The mathematical representation of these neighbors is:

$$f_l(P) = \begin{cases} f(x - \sin(l), y + \cos(l)) & : l = k\pi/2, \\ f(x - \sqrt{2}\sin(l), y + \sqrt{2}\cos(l)) & : l = k\pi/4, \end{cases} \quad (4)$$

and directional differential is defined as (5).

$$\nabla_l f(x, y) = f_l(x, y) - f(x, y), \quad (5)$$

The central difference $g_k(x, y)$ of image $f(x, y)$ on an area $[0, X]$ by $[0, Y]$ is defined as (6):

$$g_k(x, y) = \begin{cases} f(x, y) & : x = 0, X \text{ or } y = 0, Y. \\ f_l(x, y) - f_{l-\pi}(x, y) & : 0 < x < X \text{ and } 0 < y < Y. \end{cases} \quad (6)$$

where $l = k\pi/4, k \in [0, 3] : k \in Z$. Based on (5) and (6), two versions or classes of feature points were redefined [29]. The first class feature points are the extremum points in an image, if $\forall l \in L, L = \{n\pi/4 : k \in [0, 3]\}$ then $\nabla_l f(x, y) \cdot \nabla_{l-\pi} f(x, y) > 0$. and the second version or class feature points are the points whose neighbors have extrema along certain directions, if $\forall l \in L, L = \{n\pi/4 : k \in [0, 3]\}$ then $\nabla_l g_k(x, y) \cdot \nabla_{l-\pi} g_k(x, y) > 0$. The results of this process are shown in Figure 2, in Figure 2(a) we have the input the alarm clock, Figure 2(b) shows the first class feature points and in Figure 2(c) we have the second class feature.

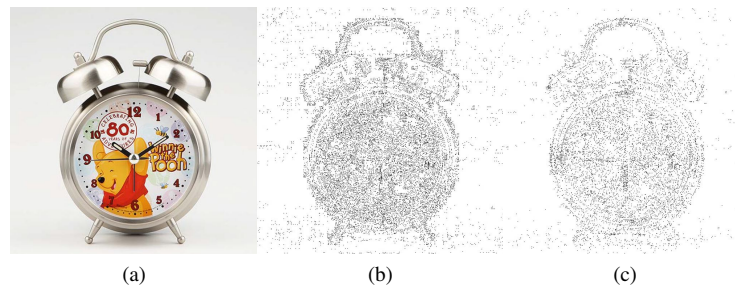


Figure 2. Representation of feature points extraction (a) input image, (b) first class feature points, and (c) second class feature points

2.2.2. Object image information measure

In gray-scale image $f(x, y)$, the information measure (IM) of the image IM is defined in the equation 7, where I_1 is the first class feature point and I_2 is the second class feature point as (7):

$$IM = I_1 - I_2, \quad (7)$$

regarding color images, we make the sum of the three channels to compute the information measure. Furthermore, IM changes are used to predict the object size in a sequence.

2.2.3. Scale updating process

The scale of the window tracking is updated every N frames according to the change of IM Figure 3 shows different IM size possibilities. The scale updating process begins by calculate the IM on the nTh current frame I_1 on the tracking window see Figure 3(a), then we calculate new IMs to get two window sizes by multiplying a factor of $1 + \alpha$ and $1 - \alpha$, then perform the same steps on the $n + Nth$ frame to get three IMs I_4, I_5 and I_6 . Thus, based on the change of the information measure within each layer, we can judge whether the size of the object is increasing like in Figure 3(b) or decreasing as shown in Figure 3(c).

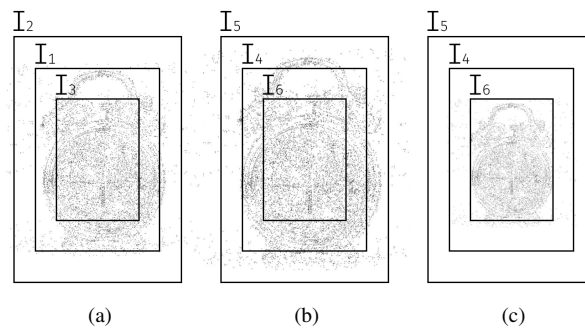


Figure 3. Representation object sizes and the information measures (a) nTh frame, (b) $n+Nth$ frame with scale increasing, and (c) $n+Nth$ frame with scale decreasing

The ratio of the information measure and the object size are related. Thus when $I_5 > I_2$, the object size had increased, and the increasing ratio S is defined as (8):

$$S = \begin{cases} \lg(\beta) \cdot \frac{I_5 - (I_2 - I_1)}{I_1} & : \frac{I_5}{I_4} > \frac{I_2}{I_1}, \\ 0 & : \text{else.} \end{cases} \quad (8)$$

and when $I_2 > I_5$ the object's scale may be decreased and the ratio S is as (9):

$$S = \begin{cases} \lg(\beta) \cdot \frac{I_4}{I_1} & : I_1 > I_4, \\ 0 & : \text{else.} \end{cases} \quad (9)$$

where β is a parameter that eliminates the influence of background. After calculating the scale change factor S of the n th frame, the dimension of the tracking window is updated according to the formula (10).

$$\begin{aligned} H &= H * (1 + S), \\ W &= W * (1 + S), \end{aligned} \quad (10)$$

2.3. Proposed method

In this work, we propose a system that based on a probabilistic algorithm. The purpose is to find a rough solution within searching space, this solution is improved with the propagation technique with a distribution. The scope of application of our proposed method, which is applied on stationary camera sequences, and the tracking process goes through the different stages.

2.3.1. Adaptive particle filter

As known particle filter is a meta-heuristic method, which looks for the best state estimate in dynamical system. It is a hypothetical detector that approximates the filtered posterior distribution by a set of weighted particles, it weights the particles based on a likelihood score and then propagates these particles according to a movement model which is explained in the next section. Based on fundamentals of PF we propose our adaptive particle filter (APF) tracker. APF relies on a deterministic search window, whose color content corresponds to histogram color model, and the likelihood weights is the result of normalized Euclidean distance between target color histogram and particle color histogram

In fact, the particle state modeling in this application case will be the location of each state within the image. The state space is represented in the spatial domain $P_i = (x, y)$, where the size of the object is fixed. After generating a set of N particles, we initialize the states randomly for the first image I_0 . Then we calculate for each particle state $P_{1, \dots, N}$ its red, green, and blue (RGB) normalized histogram $hist_{rgb}^{P_i} = [H_r^{P_i} \ H_g^{P_i} \ H_b^{P_i}]$, then, we calculate the Euclidean distance between each color channel. As a result, we end up with three distances as (11), (12), and (13):

$$d_r^{P_i} = \sqrt{\sum (Tr_r - H_r^{P_i})^2} \quad (11)$$

$$d_g^{P_i} = \sqrt{\sum (Tr_g - H_g^{P_i})^2} \quad (12)$$

$$d_b^{P_i} = \sqrt{\sum (Tr_b - H_b^{P_i})^2} \quad (13)$$

where Tr_r , Tr_g and Tr_b are the target's histogram for the R, G and B channel successively. Whereas, $d_r^{P_i}$, $d_g^{P_i}$ and $d_b^{P_i}$ are the Euclidean distances between i th particle state and target histogram. Thus, these three distance values represent the likelihood weights.

On filtering stage PF is based on weight values, we grouped the three RGB distances in one normalized vector see (14) and (15), that represent weight's vector for particular state.

$$w_{1, \dots, N}^t = \sqrt{d_r^{P_i^2} + d_g^{P_i^2} + d_b^{P_i^2}}, \quad (14)$$

$$w_{1,\dots,N}^t = \|d_{rgb}^{P_i}\|, \quad (15)$$

After computing the weight of each particle state, we minimize the set of weights, in order to deduce the state P_i^t most similar to the target for the iTh state particle at the frame I_t . Then we get, and we save its position on the frame. Based on this state position at resampling stage, we will apply our proposed distribution re-sampling (DS), which aims to generate new particles state for the next frame. The function defined by (16) and (17):

$$P_{1,\dots,N}^{t+1} = DS(p_i^t) \quad (16)$$

$$P_{1,\dots,N}^{t+1} = \{P_1^{t+1}, \dots, P_N^{t+1}\} \quad (17)$$

2.3.2. Proposed resampling distribution

On the PF resampling phase, we are generating new particles for those who are more likely to have low weights, to be used in the next generation. Our idea is to improve this resampling step, by generating new particles that could give higher weights. Consequently, we will have more chance to find the best estimate of the target position within the area of research (AOR).

Generally, if we use a normal random propagation, the propagation of the states in the space does not necessarily offer good candidates similar to the target. Therefore, we propose to use kernel distribution shape while propagating particles within AOR. There are a lot of benefits by using this approach. Firstly, we will get denser particles that will have higher weights. Secondly, we guarantee a high density of particles around the previous position of the target since the movement of the object between two successive frames is very small. Lastly, we propagate less dense particles to make sure that we have the chance to find the target when it makes a larger step motion. In this work, we used three kernels namely Gaussian, Epanechnikov and Triangular. Where values of distribution are between [0 and 1] for computational purposes and their equation definitions are as respectively as (18), (19), and (20):

$$f(x, y) = \exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \quad (18)$$

$$f(x, y) = \frac{3}{4} * \left(1 - \left(\frac{x^2}{\sigma} + \frac{y^2}{\sigma}\right)\right) \quad (19)$$

$$f(x, y) = (1 - \|x - y\|) \quad (20)$$

So, after computing the likelihood weights of all particles, then we choose among them the most similar particle candidate, where the lowest Euclidean distance we get the most similarity we have. After that we take the coordinates of the best estimate particle as input parameters into the resampling function, in order to build the next particles states for the next frame within the AOR. According to the values of the distribution, we deduce the number of fields. A field is a region in which we propagate a precise amount of particles. Figure 4 shows the general cases that could appear. Fields could have a circular like Figure 4(a) or an ellipse shape see Figure 4(b) depending on the shape of the target. Thus we have chosen elliptical shape because it suits kernel distribution, and it ensures covering the whole object with particles.

On resampling step, we apply our contribution that integrates kernel distribution, where the main steps are described as:

- Build the distribution: first, we create a distribution according to the size of the AOR and the type of kernels.
- Find the peak of the distribution : after, we get the kernel distribution result, we try to find its global Maxima and we save its position for further calculations.
- Calculate the horizontal and vertical means: based on the global Maxima, we define a vertical and a horizontal path from the peak to the null value, where these null values are located in the edges, then we retrieve these values to calculate the mean of each side.
- Based on the means, we calculate the number of fields and their width and height: in this step, we specify the perimeters of each field. Since we are working with elliptical shape, its equation needs parameters, especially the radius of the x-axis (width), where we will be using a horizontal cross-section of the kernel

as a table. Likewise, the radius of the y-axis (height) will be calculated using a vertical cross-section of the kernel as well. Figure 5 shows a horizontal cross-section of the kernel, we consider it as a table of N elements. Where the global Maxima is presented in yellow, the blue represents the edge of the kernel, and the green represents the elements between the peak of the kernel and its last null value. Firstly, we start from the global Maxima to calculate the number of elements constituting the radius of the first field. On x-axis we check if the value of the first element is greater than or equal to the horizontal means, if it is true, then this field will have a width radius of one single element, otherwise we add the next element and we check if their sum is greater than or equal to the horizontal means, if this is the case then the width of this field has two elements, else we add a third element to the two previous ones and we iterate again. We repeat this process until we find the length of the first field. Secondly, we move to the second field and we apply the same process, however, the element of the first field is not re-used, instead we start from the element following the last element of the previous field. In fact, the width of the second field is equal to the width of the first field plus the number of elements found for the second field.

- Calculate the particle's portions on each field: at this step, we try to measure how many particles inside a field. First, we calculate the sum of the cross-section of the kernel, where the direction of the cross-section has a negligible impact on the calculation. Next, we calculated for each field its percentage according to (21):

$$Pr_{field\ t} = \frac{\sum_1^{N_{field\ t}} V_i}{\sum_1^{N_{cs}} T_i} * 100 - Pr_{field\ t-1} \tag{21}$$

where $Pr_{field\ t}$ and $Pr_{field\ t-1}$ are percentages of the current field and the previous field respectively, $N_{field\ t}$ is the number of elements of the current field, N_{cs} is the number of element of the cross-section table.

- Propagate particles on each field : after finding the portion on each field, we try to build these fields using the pair of radius found previously, then propagate their portion inside them.
- Combine the results: finally, we will have several fields of particles superimposed, we try in this step to combine them in one single structure of particles distributed according to kernel norm.

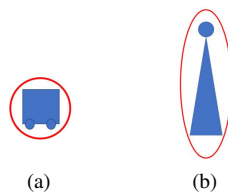


Figure 4. The target shapes (a) circular shape and (b) ellipse shape

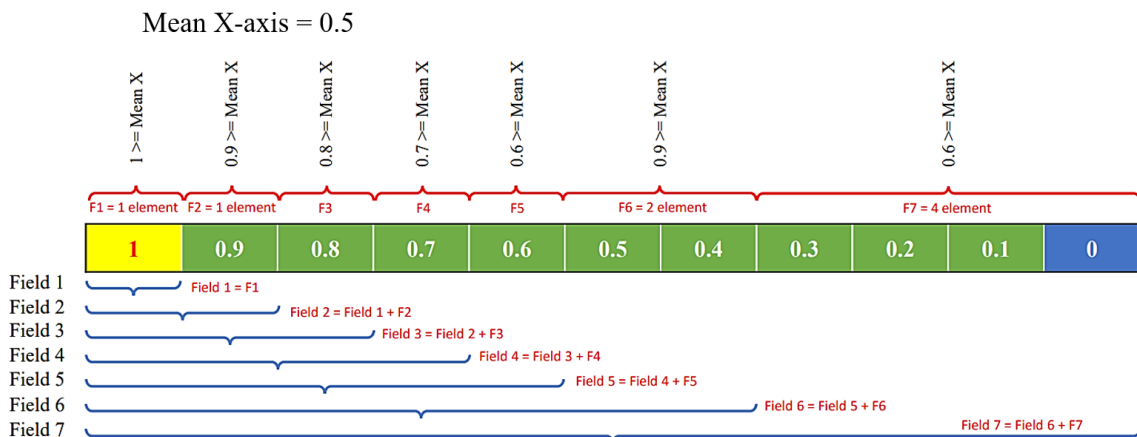


Figure 5. Overview of the field calculation process

2.3.3. Proposed algorithm: multi-scale adaptive particle filter algorithm

Many tracking systems rely on object appearance, and it's the main point that affects the performance of each one. This work presents a new approach based on PF and estimation technique to construct a framework base on statistical multi-reference histogram. The general idea of our proposed work is to build an accurate target tracking system by using adapted PF and Euclidean distance on RGB channels, these channels histograms are considered as a feature representation, and it updates the size of the tracking window according to the information measure on color image. Our proposed method contains three main steps. After initializing the parameters and locating the target by histograms of oriented laplacian (HOL) detector [32]. The first step is predicting, we use a random uniform model to give for each particle state its coordinates inside the AOR, which we specify it according to the initial location detected. In the second step, the filtering step computes the likelihood weight for every state particle based on the minimization of these weights, we pick the greatest state as a result, which will be considered as the new location of the target on the current frame. Lastly, in the resampling step, we have developed this section in order to generate more distinctive and effective particle state using a specific kernel distribution. The general process of the proposed algorithm for multi-scale object tracking is summarized in Figure 6.

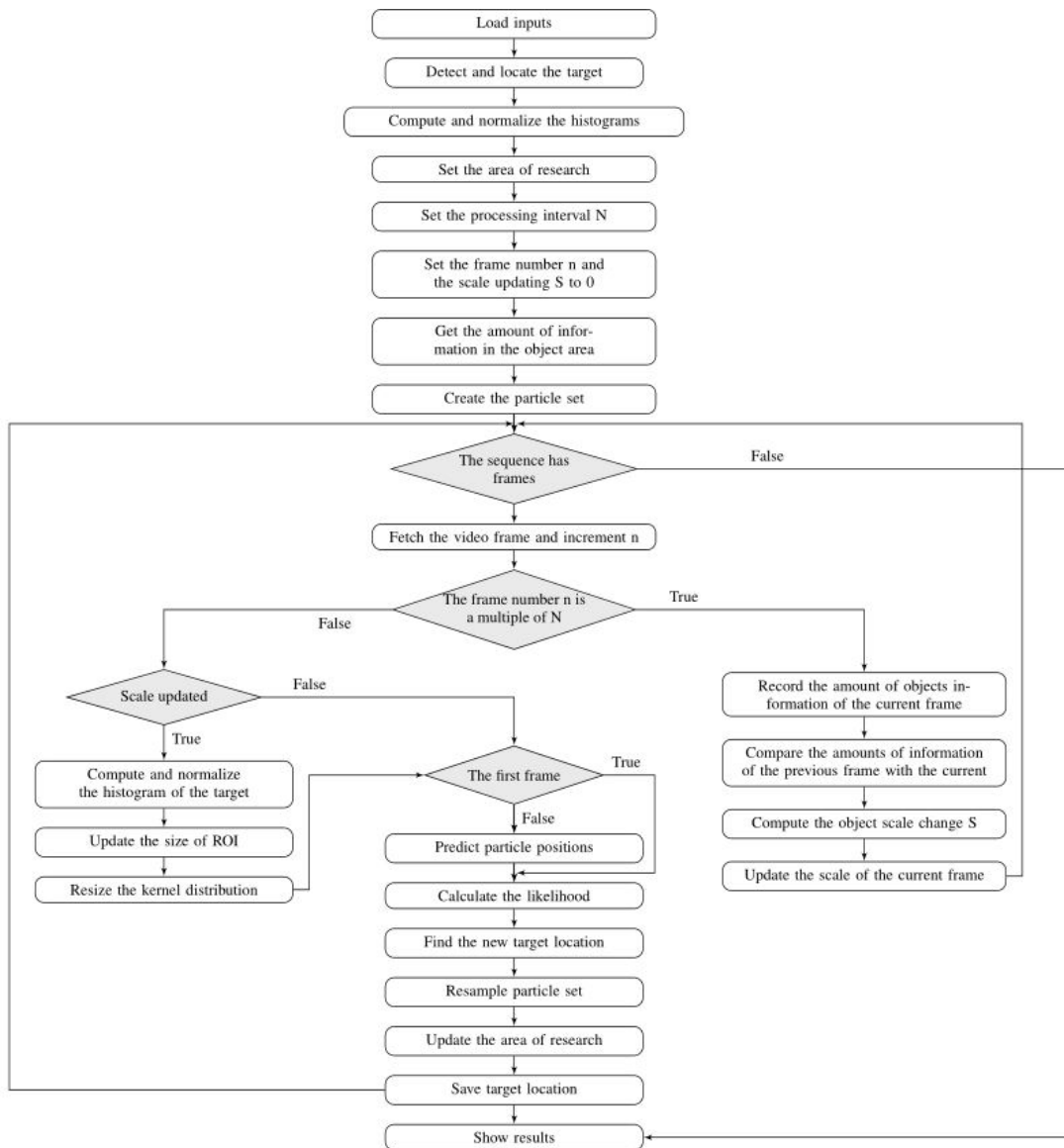


Figure 6. Overview of the proposed method

3. RESULTS AND DISCUSSION

In this section, we will show the performance of our novel MSAPF approach. In fact, we have conducted qualitative and quantitative experiments, and we have compared our results with three probabilistic methods. Namely, CS, PF, and PSO. Indeed, the experiments were implemented over an Intel Core i5-2450M CPU with 2.50 GHz and RAM 1600 MHz 12GB DDR3. For experiment purposes, object detection is done manually by marking a bounding box within the first image. Then, our algorithm runs iteratively to find the best object matching region in every new frame.

3.1. Dataset

We evaluate our proposed approach on several challenging video sequences from two dataset for tracking. Namely, OTB50, OTB100 [33], and LITIV [34]. The dataset are used for object tracking benchmark and they are frequently processed by researchers in computer vision tracking field. Obviously, these datasets guarantee a heterogeneous environment and background on which the tests will be conducted. For instance, OTB dataset provides a wide range of analyses and depicts a detailed view of tracker performance. Overall, the sequences used can group two or more difficulties such as illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, out-of-view, in-plane rotation, out-of-plane rotation, background clutters, and low resolution. The ground truth position has been used in target initialization in the first frame for each compared algorithm.

3.2. Evaluation

The goal of the following evaluation is to ensure an impartial comparison, where our approach and the popular state of the art methods have undergone several tests. These tests are presented in detail at this section, and their results are mentioned in Tables 1, 2, Figure 7, and at Figure 8. One of the most used quantitative performance evaluation is center location error (CLE) [35], this criteria allow to calculate the localization errors between the center of the tracking results and the ground truth annotations.

$$CLE = \frac{1}{N_{gr}} * \sum_{i=0}^{N_{gr}} \sqrt{(X_{gi} - X_{ri})^2 + (Y_{gi} - Y_{ri})^2}, \quad (22)$$

Where N_{gr} is the total number of images for each frame, (X_{gi}, Y_{gi}) is the position of the ground truth, and (X_{ri}, Y_{ri}) is the tracking result at frame I_i , respectively. According to the (22), the smaller CLE we get, the better optimal algorithm is. Moreover, as defined in (22), the size of the bounding boxes is neglected by this quantitative metric.

Furthermore, we also use the overlap ratio (OR) [35] to perform evaluation, it measures the success rate, and it also gives an idea on how our proposal performers over sequences in which the object changes its size. The OR is defined by (23):

$$OR_i = \frac{area(R_i \cap G_i)}{area(R_i \cup G_i)}, \quad (23)$$

With R_i is the tracked bounding box, G_i is the ground truth bounding box, where $R_i \cap G_i$ represents the intersection, and $R_i \cup G_i$ is the union of two regions. We consider the tracking has succeeded, if the $OR \geq 0.5$.

Moreover, we get the success ratio (SR) by setting an overlap score r which is defined as the minimum overlap ratio, which can decide whether an output bounding box is correct or not, it is calculated by (24):

$$SR = \sum_{i=0}^N \frac{u_i}{N_{gr}}, SR \in [0, 1], \quad (24)$$

$$u_i = \begin{cases} 1 & \text{if } OR_i \geq r \\ 0 & \text{if not,} \end{cases}$$

where N is the frame number, the decision of a corresponding tracking result is good or bad depending on the overlap ratio threshold r . As defined in equation 24, we calculate the percentage of success location according to the threshold. If SR achieves maximum score and OR is higher in this case the tracker performs great. Otherwise, it performs the worst [36].

3.3. Result analysis

In the following section, we discuss the results of the metrics that we cited above. The first evaluation was made on 17 sequences, we have made multiple tests with three distribution kernels (Epanechnikov, Gaussian and Triangular) to see their influence on the accuracy, with different particles set sizes between 500 and 800, see Table 3. The reason that we have chosen that interval, is the unsatisfying result when the number of particles is less than 500 and when we increase the number it does not give a convincing accuracy. As a result, the overall performance evaluation demonstrates that the APF method, which implements Triangular kernel with 600 particles, outperforms all the compared APF with different parameters [37].

Table 1. Average errors for target localization (by pixel)

| Sequence | MSAPF | APFT600 | PF | CS | PSO | DS |
|-----------|--------------|---------|--------|--------|-------|--------|
| BlurCar3 | 5.19 | 24.99 | 26.63 | 73.33 | 75.86 | 258.67 |
| BlurFace | 9.73 | 19.57 | 21.35 | 26.15 | 31.30 | 272.43 |
| BlurOwl | 105.68 | 14.04 | 140.07 | 27.12 | 24.84 | 370.90 |
| CarScale | 31.42 | 27.22 | 31.63 | 59.05 | 22.33 | 4.98 |
| Dog | 8.88 | 28.89 | 31.21 | 61.33 | 29.56 | 50.45 |
| Doll | 5.16 | 12.58 | 20.36 | 65.74 | 29.56 | 210.61 |
| Dragon | 20.62 | 20.86 | 33.56 | 20.52 | 14.23 | 107.83 |
| FaceOcc1 | 20.38 | 18.17 | 18.05 | 35.18 | 27.95 | 86.73 |
| Football1 | 5.40 | 27.18 | 32.51 | 59.95 | 26.89 | 155.02 |
| Girl | 5.46 | 18.99 | 16.57 | 11.56 | 27.93 | 47.51 |
| Gym | 10.92 | 10.67 | 10.98 | 22.09 | 19.79 | 47.57 |
| Jogging | 11.50 | 11.27 | 11.76 | 22.37 | 18.49 | 12.82 |
| jp1 | 5.31 | 28.75 | 29.55 | 38.63 | 11.72 | 155.14 |
| jp2 | 21.01 | 18.33 | 57.06 | 50.29 | 7.05 | 107.31 |
| Skater | 13.50 | 12.73 | 12.67 | 22.92 | 17.10 | 21.51 |
| Surfer | 26.90 | 23.15 | 188.79 | 131.32 | 24.35 | 29.34 |
| wbook | 8.92 | 23.63 | 24.03 | 20.31 | 23.45 | 130.44 |
| Mean | 18.59 | 20.06 | 41.57 | 43.99 | 25.43 | 121.72 |

Table 2. Average success rate (%) tracking)

| Sequence | MSAPF | APFT600 | PF | CS | PSO | DS |
|-----------|--------------|---------|-------|-------|-------|-------|
| BlurCar3 | 98.9 | 57.42 | 54.34 | 24.09 | 22.41 | 15.97 |
| BlurFace | 98.86 | 82.76 | 60.93 | 67.75 | 17.16 | 0 |
| BlurOwl | 7.45 | 85.42 | 44.69 | 51.35 | 65.77 | 0 |
| CarScale | 82.14 | 35.71 | 34.92 | 14.68 | 24.6 | 99.6 |
| Dog | 83.78 | 16.54 | 14.96 | 6.3 | 6.6 | 3.15 |
| Doll | 98.1 | 52.27 | 50.54 | 35.38 | 60.46 | 0 |
| Dragon | 87.97 | 52.21 | 48.67 | 50.44 | 68.14 | 0 |
| FaceOcc1 | 97.3 | 99.55 | 98.88 | 69.62 | 90.36 | 0 |
| Football1 | 77.03 | 25.68 | 16.22 | 9.46 | 14.86 | 0 |
| Girl | 90.2 | 51.2 | 29.8 | 53.6 | 14.6 | 0 |
| Gym | 75.12 | 31.29 | 31.81 | 25.55 | 31.03 | 78.23 |
| Jogging | 66.45 | 67.1 | 66.45 | 52.12 | 54.07 | 94.79 |
| jp1 | 99.1 | 46.71 | 45.23 | 66.12 | 85.36 | 0 |
| jp2 | 65.46 | 72.93 | 53.71 | 48.03 | 95.2 | 0 |
| Skater | 81.75 | 78.13 | 78.75 | 65 | 79.38 | 66.25 |
| Surfer | 52.66 | 28.46 | 3.72 | 6.91 | 21.28 | 0 |
| wbook | 98 | 73.32 | 71.94 | 74.18 | 63.34 | 0 |
| Mean | 80.02 | 56.28 | 47.39 | 42.39 | 47.92 | 21.05 |

The next evaluation was conducted on on multiple tracking algorithms. Namely, PF, PSO, CS, APF with triangular kernel and 600 particles (APFT600), MSAPF use also with triangular kernel and 600 particles and Deep sort (DS). Where Table 1 demonstrates the average CLE and Table 2 shows the average SR. The experiment results are highlighted within the dashed box, in which our proposed approach MSAPF gets good results. It gives 18.59 pixels in average CLE and it has the highest average SR of 80.01%, followed by APFT600 in the second place with performance of 56.28% in average SR and an average CLE of 20.06 pixels. Then comes PSO in the third place. While PF and CS algorithm takes the next place. The last rank is taken by the DS algorithm.

Table 3. Kernel type, average precision (%) and tracking speed (fps) of our method under different particles number

| Kernel | Epanechnikov | | | | Gaussian | | | Triangular | | | | |
|--------------|--------------|-------|-------|------|----------|-------|-------|------------|-------|-------|------|------|
| Nb particles | 500 | 600 | 700 | 800 | 500 | 600 | 700 | 800 | 500 | 600 | 700 | 800 |
| Speed | 15.32 | 11.43 | 10.28 | 6.36 | 14.19 | 10.17 | 8.67 | 7.18 | 16.09 | 11.15 | 9.06 | 7.73 |
| Precision | 80.07 | 80.44 | 82.36 | 80.9 | 81.03 | 82.37 | 81.06 | 80.99 | 79.91 | 82.63 | 82.4 | 80.6 |

To compare the tracking results in details between our proposed approach and the existing trackers methods. We highlight the overlap ratio plots in Figure 7 where we have calculated OR plots namely for wbook sequence in Figure 7(a), blurcar3 sequence in Figure 7(b) and blurface sequence in Figure 7(c). As well as the success ratio plots Figure 8, where we have calculated SR plots for same sequences, wbook in Figure 8(a), wbook in Figure 8(b) and wbook in Figure 8(c). This is shown over 3 sequences. Apparently, from the Figure 7, we can see that our tracker maintains a higher and stable overlap score along the sequences. Additionally, we use the success ratio plot to figure out the average performance of trackers on every sequence. The success plot gives the percentage of frames where the overlap ratio is higher than a threshold. As illustrated in Figure 8 our improved MSAPF approach achieves the highest score in terms of this evaluation versus the state of the art trackers. The above analysis implies that our approach performs more accurate and gives stable results than the other trackers. We can say that our tracker illustrates the expectations on the results.

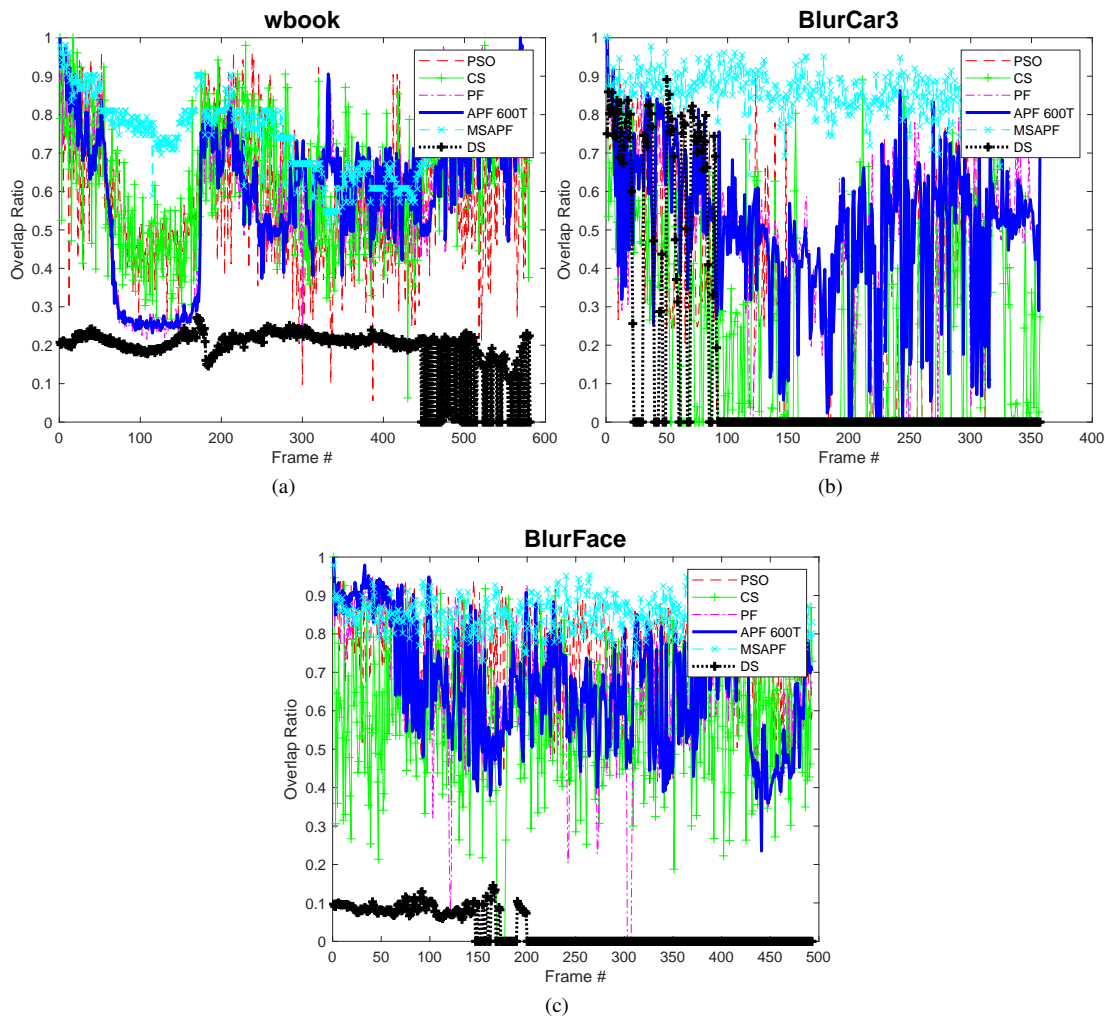


Figure 7. Overlap ratio plots of the trackers, (a) wbook sequence, (b) blur car 3 sequence and (c) blur face sequence

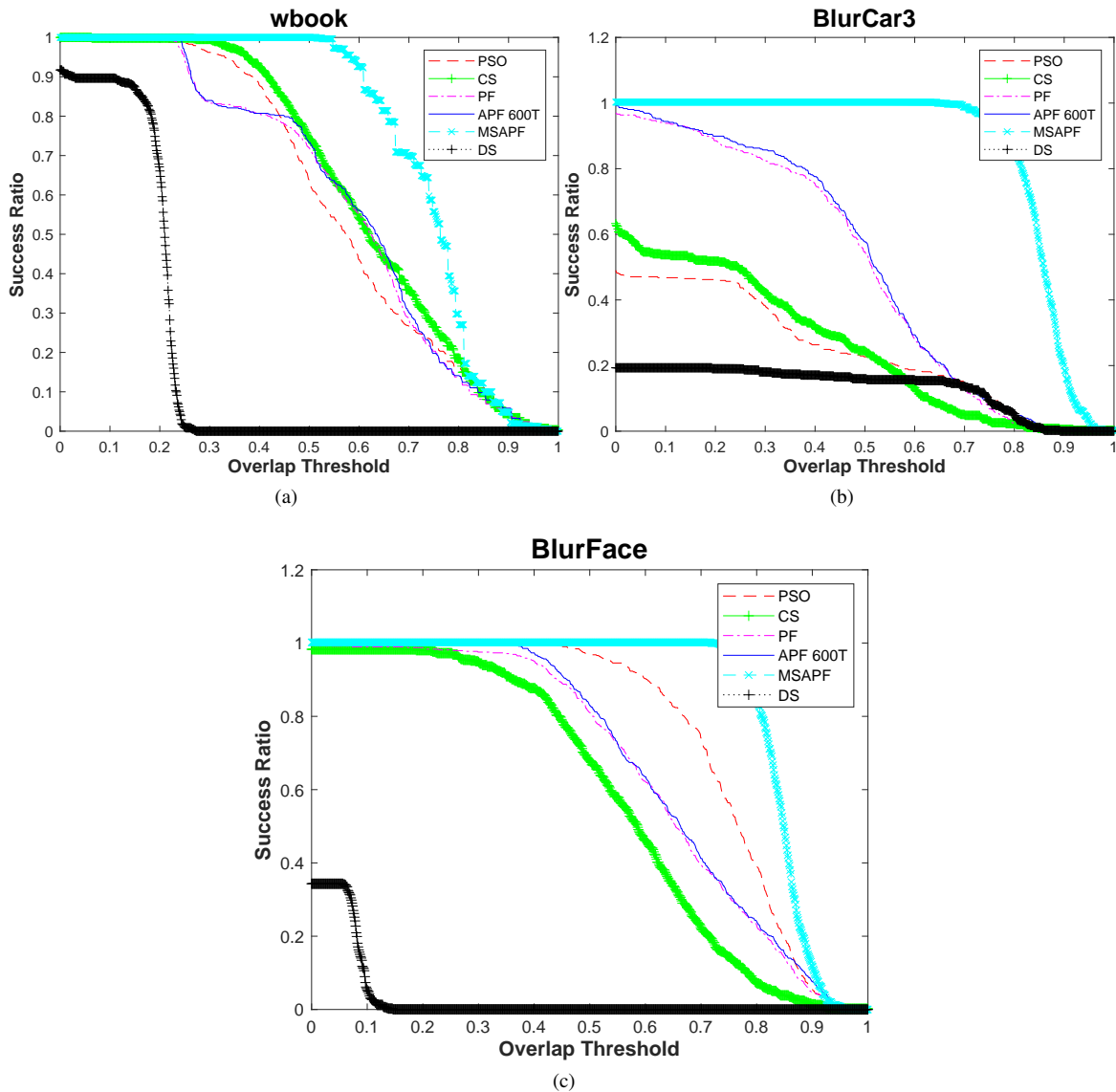


Figure 8. Success ratio plots of the trackers, (a) wbook sequence, (b) blur car 3 sequence, and (c) blur face sequence

4. CONCLUSION

The deterministic approaches in the literature have proven their efficiency in many tracking cases, however, there are scenarios where they showed their negative points especially at the time running, resource use and on accuracy rate. In this paper, a novel object tracker approach has been proposed by integrating kernel distribution inside PF probabilistic algorithm with scale updated tracking window, in order to give more intense candidate around the region of interest and the size of the tracking window changes according to the object information measure of the object. This feature has improved our previous tracker APF based in triangular kernel and the size $n=600$ particle significantly. Furthermore, MSAPF is compared with three state-of-art algorithms especially PSO, PF, CS and DS, where the evaluation was conducted on two benchmark datasets OTB and LITIV, and the examination results show that the success rate of MSAPF tracker outperforms the APFT600 based on fixed tracking window, PSO, PF, CS, and DS and the target localization error is much lower than these algorithms.

ACKNOWLEDGEMENT

This work is supported by CNRST (Centre National de la Recherche Scientifique et Technique). Ministry of Higher Education and Scientific Research Morocco, and by the "SAFEROAD: Métaplateforme de la Sécurité Routière" project under Contract No: 24/2017.





REFERENCES

- [1] P.-H. Yuan, K.-F. Yang, and W.-H. Tsai, "Security monitoring around a video surveillance car with a pair of two-camera omni-directional imaging devices," in *International Computer Symposium (ICS2010)*, Dec. 2010, pp. 325–330, doi: 10.1109/COMPSYM.2010.5685495.
- [2] V. C. Banu, I. M. Costea, F. C. Nemanu, and I. Badescu, "Intelligent video surveillance system," in *IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Oct. 2017, pp. 208–212, doi: 10.1109/SIITME.2017.8259891.
- [3] Q. Feng, C. Gao, L. Wang, M. Zhang, L. Du, and S. Qin, "Fall detection based on motion history image and histogram of oriented gradient feature," in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov. 2017, pp. 341–346, doi: 10.1109/ISPACS.2017.8266500.
- [4] S. Tayb, Y. Azdoud, A. Amine, B. Nassih, H. Hachimi, and N. Hmina, "HOL, GDCT and LDCT for pedestrian detection," *Computer Science and Information Technology (CSIT)*, Jan. 2016, pp. 87–96, doi: 10.5121/csit.2016.60109.
- [5] S. B. Setyawan, H. Arrosida, A. Sudaryanto, and R. G. P. Yudha, "Robust face tracking using improved mean shift algorithm," *Journal of Physics: Conference Series*, vol. 1845, no. 1, Mar. 2021, doi: 10.1088/1742-6596/1845/1/012055.
- [6] F. Bousetouane, L. Dib, and H. Snoussi, "Improved mean shift integrating texture and color features for robust real time object tracking," *The Visual Computer*, vol. 29, no. 3, pp. 155–170, Mar. 2013, doi: 10.1007/s00371-012-0677-0.
- [7] S. Jansen, A. Shantia, and M. A. Wiering, "The neural-SIFT feature descriptor for visual vocabulary object recognition," in *International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, pp. 1–8, doi: 10.1109/IJCNN.2015.7280660.
- [8] I. Leichter, "Mean shift trackers with cross-bin metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 695–706, Apr. 2012, doi: 10.1109/TPAMI.2011.167.
- [9] Z. Qin and C. R. Shelton, "Improving multi-target tracking via social grouping," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 1972–1978, doi: 10.1109/CVPR.2012.6247899.
- [10] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Systems*, vol. 195, May 2020, doi: 10.1016/j.knsys.2020.105697.
- [11] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2021, doi: 10.1109/TIP.2020.3037518.
- [12] D. Yuan, X. Shu, and Z. He, "TRBACF: learning temporal regularized correlation filters for high performance on-line visual object tracking," *Journal of Visual Communication and Image Representation*, vol. 72, Oct. 2020, doi: 10.1016/j.jvcir.2020.102882.
- [13] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 983–990, doi: 10.1109/CVPRW.2009.5206737.
- [14] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 49–56, doi: 10.1109/CVPR.2010.5540231.
- [15] M. Maria, S. El Kebir, and H. Ahmed, "Reduction dimensionality of hyperspectral imagery using genetic algorithm and mutual information and normalized mutual information as a fitness function," *International Review of Applied Sciences and Engineering*, vol. 12, no. 1, pp. 64–75, Mar. 2021, doi: 10.1556/1848.2020.00149.
- [16] M.-L. Gao, L.-J. Yin, G.-F. Zou, H.-T. Li, and W. Liu, "Visual tracking method based on cuckoo search algorithm," *Optical Engineering*, vol. 54, no. 7, Jul. 2015, doi: 10.1117/1.OE.54.7.073105.
- [17] H. Nenavath, K. Ashwini, R. K. Jatoth, and S. Mirjalili, "Intelligent trigonometric particle filter for visual tracking," *ISA Transactions*, Sep. 2021, doi: 10.1016/j.isatra.2021.09.014.
- [18] K. Ashwini, M. R. Devi, and H. Nenavath, "Comprehensive learning particle swarm optimizer for intelligent visual tracking," in *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Feb. 2021, pp. 1112–1119, doi: 10.1109/ICICV50876.2021.9388501.
- [19] H. Yu, A. Sharma, and P. Sharma, "Adaptive strategy for sports video moving target detection and tracking technology based on mean shift algorithm," *International Journal of System Assurance Engineering and Management*, pp. 1–11, May 2021, doi: 10.1007/s13198-021-01128-5.
- [20] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, Jun. 2019, doi: 10.1007/s11042-018-6800-0.
- [21] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, 2014, doi: 10.5244/C.28.65.





- [22] D. Yuan, X. Li, Z. He, Q. Liu, and S. Lu, "Visual object tracking with adaptive structural convolutional network," *Knowledge-Based Systems*, vol. 194, Apr. 2020, doi: 10.1016/j.knsys.2020.105554.
- [23] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6931–6939, doi: 10.1109/CVPR.2017.733.
- [24] V. Mandal and Y. Adu-Gyamfi, "Object detection and tracking algorithms for vehicle counting: a comparative analysis," *Journal of Big Data Analytics in Transportation*, vol. 2, no. 3, pp. 251–261, Dec. 2020, doi: 10.1007/s42421-020-00025-w.
- [25] M. A. B. Zuraimi and F. H. K. Zaman, "Vehicle detection and tracking using YOLO and DeepSORT," in *2021 IEEE 11th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, Apr. 2021, pp. 23–29, doi: 10.1109/ISCAIE51753.2021.9431784.
- [26] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7575, no. 4, 2012, pp. 702–715.
- [27] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van De Weijer, "Adaptive color attributes for real-time visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1090–1097, doi: 10.1109/CVPR.2014.143.
- [28] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, Mar. 2015, doi: 10.1109/TPAMI.2014.2345390.
- [29] H. Qian, Y. Mao, J. Geng, and Z. Wang, "Object tracking with self-updating tracking window," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4430, 2007, pp. 82–93.
- [30] A. Lindhe, C. Ringqvist, and H. Hult, "Particle filter bridge interpolation," *arXiv:2103.14963*, Mar. 2021.
- [31] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*, 4th ed. Cengage Learning, 2013.
- [32] Y. Azdoud, A. Amine, N. Alioua, and M. Rziza, "Pre collision detection system for pedestrian safety based on HOL," in *IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, Nov. 2015, pp. 1–2, doi: 10.1109/AICCSA.2015.7507137.
- [33] benchmark, "Visual-tracker Google groups." <http://www.visual-tracking.net/> (accessed Dec. 10, 2020).
- [34] "Laboratoire d'interprétation et de traitement d'images et vidéo (LITIV) codes and datasets," Polytechnique Montréal's. accessed Dec. 10, 2020. [Online]. Available: <http://www.polymtl.ca/litiv/en/codes-and-datasets/>.
- [35] W. Tang, B. Liu, and N. Yu, "Deep scale feature for visual tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10666, Springer, Cham, 2017, pp. 306–315.
- [36] K. Hu, E. Fan, J. Ye, C. Fan, S. Shen, and Y. Gu, "Neutrosophic similarity score based weighted histogram for robust mean-shift tracking," *Information*, vol. 8, no. 4, Oct. 2017, doi: 10.3390/info8040122.
- [37] Y. Azdoud, A. Amine, and H. Hachimi, "Adaptive particle filter tracker combining kernel distribution to particle filtering to better enhance object tracking," *International Journal of Imaging and Robotics*, vol. 20, no. 4, 2020.

BIOGRAPHIES OF AUTHORS







Azdoud Youssef     Ph.D. student in Computer Engineering, affiliated to the Systems Engineering Laboratory in the National School of Applied Sciences (ENSA), Ibn Tofail University, Kenitra, Morocco. He received his master's degree in computer engineering: Master of Information Systems Security from the National School of Applied Sciences of Kenitra, Ibn Tofail University, Kenitra, Morocco. He is currently working in the field of computer vision. He can be contacted at email: youssef.azdoud@uit.ac.ma.







Amine Aouatif     has been an associate professor in the Ibn Tofail University - Kenitra, Morocco, since 2014. In July 2010, she joined the ENSA of Kenitra. Ex Vice President of IEEE SPS Morocco Chapter, Secretary General of the Moroccan Association for the Development of Electronics, Electrical Engineering, Computer Science and Automation (AMADEIA). Her areas of research include data classification, object tracking, road safety for intelligent vehicles and pattern recognition. She can be contacted at email: aouatif.amine@uit.ac.ma.



Nassih Bouchra     received her Ph.D. degree in 2018 for a dissertation titled at Ibn Tofail University, in National School of Applied Sciences ENSA Kenitra, Morocco, she received the B.S. degree in Physical Science from Quods College in 2005 and the Master degree in Telecommunications and Networks from University Chouaib Doukkali, Morocco in 2011. Her main areas of research interest are including image processing, machine learning, computer vision and pattern recognition. She can be contacted at email: youssef.azdoud@uit.ac.ma.



Ngadi Mohammed     Ph.D. degree in 2018 for a dissertation titled: Apprentissage Artificiel et Reduction de Dimension:Application au Diagnostic Medical, from the Ibn Tofail University, in the National School of Applied Sciences ENSA in Kenitra. He got a Master Degree in Industrial data processing from the University of Sidi Mohammed Ben Abdellah in 2009. The main subject of his Ph.D. Project include topics such as: Object classification, medical diagnosis, feature extraction, dimensionality reduction, machine learnnig, image processing and computer vision. He can be contacted at email: Ngadi.mohammed@univ-ibntofail.ac.ma.