Enhanced object tracking with artificial bee colony, motion modeling, and deep learning

Ramdane Taglout[1,2], **Bilal Saoud**[3,4]
[1]LIM Laboratory, Faculty of Exact Sciences, University of Bouira, Bouira, Algeria
[2]Department of Information Technology, Faculty of Exact Sciences, University of Bouira, Bouira, Algeria
[3]Department of Computer Science, Mohamed El Bachir El Ibrahimi University of Bordj Bou Arreridj, Bordj Bou Arreridj, Algeria
[4]LISEA Laboratory, Faculty of Applied Sciences, University of Bouira, Bouira, Algeria

Article Info

Article history:

Received Jan 21, 2025
Revised Oct 28, 2025
Accepted Nov 8, 2025

Keywords:

Artificial bee colony
Confidence information
Correlation filter
Kalman filter
Object tracking

ABSTRACT

As a fundamental aspect of computer vision, visual object tracking supports a wide array of applications, notably in transport infrastructure and advanced industrial automation. Although correlation filter-based trackers demonstrate robust performance, they face persistent limitations including scale changes, object occlusion, boundary artifacts, and complex background interference. To address these issues, we have introduced an approach that combines artificial bee colony (ABC) optimization, deep neural architectures, and Kalman filtering techniques. Our methodology begins with reliability assessment of the tracking pipeline, proceeding to compute target confidence measures at the predicted position, followed by an adaptive update mechanism. The proposed system leverages ABC optimization for dynamic scale adaptation while employing Kalman filtering to model inter-frame target motion dynamics. Comprehensive evaluation across multiple benchmark datasets demonstrates our method's efficacy, precision, and resilience, achieving enhanced performance relative to existing state-of-the-art approaches.

Corresponding Author:

Ramdane Taglout
LIM Laboratory, Faculty of Exact Sciences, University of Bouira
Bouira, Algeria
Email: r.taglout@univ-bouira.dz

1. INTRODUCTION

Computer vision plays an important role today. Its importance lays in the problems which can solve. It can be seen as a tool that allows the digital world of an image to interact with its physical world. Computer vision has been used to solve many problems like self-driving cars [1], facial recognition [2], augmented and mixed reality [3], health care and health technology [4], and internet of things (IoT) [5]. Furthermore, computer vision is a vast domain of research with many axes. Among these axes of research, we find visual object tracking.

The major goal of this axe of research is to estimate the location of target in all frames based on the initial frame target. In other words, after identifying a target in a video frame, it is advantageous to follow its position in the next frames. Every frame where the target is correctly followed produces additional details about the target identity and motion. Nowadays, visual object tracking is a hot subject of research, where many works have been published [6]. These works aim to improve the existing methods and overcome challenging scenarios such difficult situations including alterations in the target appearance and tracking targets through intricate movements. In addition, methods for visual object tracking are typically categorized into many classes.

They are capable of being categorized as multiple-object vs. single-object tracking methods, discriminative vs. generative methods, offline methods vs. online learning methods, methods based on non-aware or context-aware. Single object tracking methods aim to track one object at each sequence in the other hand, multi-object tracking methods can track multiple targets simultaneously [7], [8]. For generative methods, the tracking operation is completed by finding the most suitable window [9], [10]. However, discriminative methods separate the target object patch from its background [11].

Discriminative correlation filter (CF)-based trackers continue to present significant challenges due to a variety of factors. Building a robust tracker requires the development of highly discriminative features and a strong classifier. Most current DCF-based methods struggle to maintain long-term consistency, often losing critical target information from the initial frame, which leads to drift and reduced tracking reliability over time. Additionally, many trackers lack robust mechanisms for handling challenging scenarios such as occlusion, deformation, and rapid motion, resulting in decreased performance under real-world conditions.

Another significant shortcoming is the absence of effective reliability monitoring during model training, which can allow noisy or invalid data to degrade model accuracy. Furthermore, the scale filter's fixed, predefined scale range or redundant scale assumptions limit its ability to adapt to dynamic changes in target size. These limitations highlight the need for innovative solutions that can ensure information retention, robust trajectory modeling, reliable training processes, and adaptive scale management.

To overcome these issues, we have proposed a new method of visual object tracking method utilizing a CF framework, enhanced by integrating a deep learning network. Our method is shown to outperform several leading state-of-the-art approaches in experimental evaluations. The main contributions of our work are as follows:

- Using a reliable online training method that can keep important information from the first frame.
- Utilizing the Kalman filtering approach to gather target trajectory information so that the tracker can conduct robust tracking even when the object is occluded, deformable, or moving quickly.
- Using a mechanism for monitoring the process reliability during the model training guarantees that the target information is valid.
- We propose a method for assessing scale variation in target tracking algorithms that takes advantage of an optimization strategy, aiming to address redundancy and fixed scale issues commonly found in current techniques.

The paper will be structured as follows: the second section will cover our proposed method. We shall outline the method in the third part. Results and discussion will be illustrated in the fourth section. The fifth section will describe our conclusion and future directions.

2. PROPOSED METHOD

This part of the paper presents both the design principles and the operational process of the proposed object tracking approach. First of all, the efficient online training strategy, which keeps the valuable information of the first frame target intact throughout the training, will be illustrated. Then, the simulation of object motion based on Kalman filter (KF) in order to get information about its trajectory is going to be presented. Finally, the artificial bee colony optimization (ABC) will be used to enhance the precision of our tracker. The proposed tracking method involves initializing the CF with target information from the initial frame, using both the KF and the CF to estimate the target position in consecutive frames, and assessing consistency through a reliability analysis module. After achieving accurate localization, the ABC method is applied to estimate the target scale.

Next, the first frame target information and the current accurate position are used to train the model simultaneously. The CF is updated to include "r," which represents the reliability of the tracking result, and the threshold "Thr" is used to measure tracking robustness. CNN is employed for extracting visual information from the images. The ABC method is then used for target scale estimation. The framework of our new object tracker is represented in Figure 1.

2.1. Trajectory estimation based on Kalman filter

KF can provide an estimation of some unknown variables based on some measurements taken overtime. KF has been very useful in different applications. Among these applications, we find video tracking. KF uses simple forms and does not need a lot of computational power [12]. KF can be seen as an ideal filter, because it minimizes the difference between the real and the estimated state. Because of simplicity and efficiency of KF to represent target motion, a constant velocity of model has been used. This procedure can be divided into two stages, which are prediction and correction.

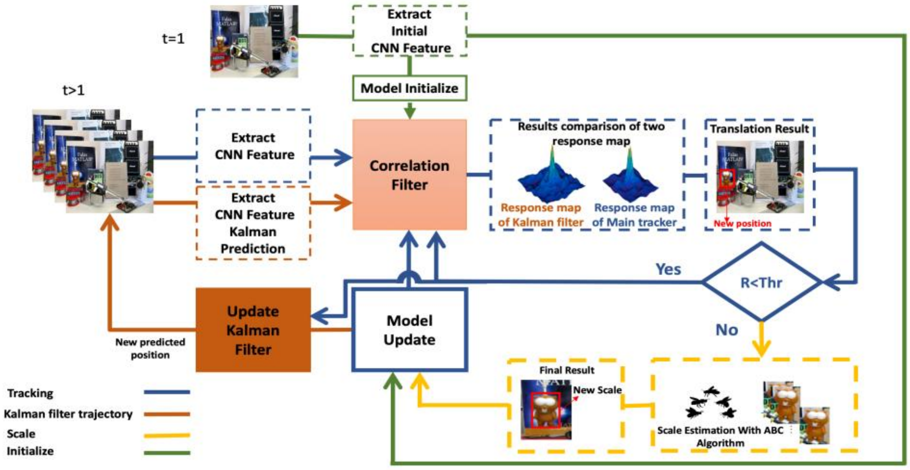


Figure 1. The framework of the proposed object tracker

2.2. Convolution target location

Standard trackers based on correlation filtering [13] have a fundamental idea, which is to learn a discriminant classifier at first, then search for the largest value of the relevant response graph to determine the estimated location of the target object. Furthermore, these trackers begin by training the filter with labels and samples. The target is then located in the search patch using the filter. Finally, the new object position will be used to update the filter. The model of this process is as (1):

$$w^* = argmin \sum \left\| w.x_{m,n} - y(m,n) \right\|^2 + \lambda \|w\|_2^2 \tag{1}$$

Where x stands for samples, w for the learned CF, y for the regression response with a Gaussian shape, λ is a parameter of regularization to overcome the over-fitting problem.

The sensitivity of deep architectures to appearance changes necessitates a robust feature extraction strategy. Consequently, we employ the pre-trained VGG Net [14] to generate a multi-layer feature representation from layers conv3-4, conv4-4, and conv5-4. The resulting feature vector x, with dimensions M (width), N (height), and D (channels). It is integrated into our tracking framework. The final output is produced by computing an inner product through the application of a linear kernel in Hilbert space, such as (2):

$$w.x_{m,n} = \sum_{d=1}^{D} w_{m,n,d}^{T} \cdot x_{m,n,d} \tag{2}$$

The model employs an M×N CF. Training data is generated by creating circular shifts of a base sample x_l, producing a set of samples $x_{m,n}$ indexed over $(m,n) \in [0,1,\dots,M-1] \times [0,1,\dots,N-1]$. Each circular sample is assigned a two-dimensional Gaussian function as (3).

$$y(m,n) = e^{-\frac{(m-M/2)^2+(n-N/2)^2}{2\sigma^2}} \tag{3}$$

The variables M and N are used to describe the dimensions of the convolutional feature map, while σ denotes the width of the kernel. Additionally, both x and the correlation filters w have identical sizes. The solution of the filter W^d can be obtained in the frequency domain, following the same procedure as depicted in [15], by applying the fast Fourier transform (FFT) to each channel. The corresponding filter in the frequency domain can be represented as (4):

$$W^d = \frac{Y \odot \overline{X}^d}{\sum_{i=1}^{D} X^i \odot \overline{X}^i + \lambda} \tag{4}$$

Complex conjugates are shown by the bars above the variables. The operator $\odot$ represents a Hadamard (element-wise) product and $d \in 1,..,D$ is the number of channels. The subsequent frame will be utilized for obtaining the ROI convolutional features. Where the first layer is specified as $Z_t \in R^{M,N,D}$. The following formula (5) shows the calculation of first layer response map Rp_l:

$$Rp_l = F^{-1}\left(\sum_{d=1}^{D} W^d \odot \overline{Z}^d\right) \tag{5}$$

By identifying the maximum value within the response map, we could find the predicted position of the target, which is provided by the first layer. We take a critical look at the response maps produced by a variety of different feature layers. To be more specific, we employ weights to combine the response maps made up of convolutional layers. To establish the definite target location, we commence by utilizing the coordinates of the highest value inside the factored composite response map as (6) to (8):

$$Rp_{fusion} = \sum_{l=1}^{3} a_l Rp_l \tag{6}$$

$$(x, y) = arg_{m,n} max Rp_{fusion}(m, n) \tag{7}$$

$$pos = (x, y) \tag{8}$$

The fusion weight of each layer is represented by a_l. We assume that the weights of the convolution layers are 1, 0.5, and 0.2 from coarse to fine, respectively. pos is the target location calculated by our tracker.

2.3. Appearance model update

As described in [16], when tracking in a complex environment, the object being tracked's information remains relatively consistent between consecutive frames and includes a significant amount of redundant information. The change and redundant information lead to slow down the tracking speed and also weaken the tracker's performance. To overcome this problem, we have used a strategy based on sparse updating in conjunction with a reliability analysis of target information. In order to estimate the reliability, we have exploited the (9) and (10):

$$\rho_l = 1 - min(\frac{r_{mean}}{r_{max}}, 0.6) \tag{9}$$

$$r_l = min(\frac{\mu_{mean}}{\mu_{max}} . \rho_l, 1) \tag{10}$$

Where r_{max} and r_{mean} represent the response map maximum and average values, respectively. This process allows us to obtain more robust and accurate filters. CF has been updated by a moving average approach. This update improves performance of tracker by avoiding significant changes in the model. The updating is as (11):

$$\widehat{W}_i^t = (1 - \eta)\widehat{W}_i^{t-1} + \eta\widehat{W}_i^{new} \tag{11}$$

The relevant learning rate is identified by η.

2.4. Scale estimation

Determining the intended size of the object in the image is critical. Among the applications, where the estimation scale is important, we find robotics and surveillance. In order to estimate the target position, the CF for locating the target is trained by extracting depth information from the image or frame. However, the scale of the target is maintained as the same as it is in the previous frame. The training of a scale filter is therefore required in order to obtain an accurate estimation of the present target scaling.

For each frame, several scales are used to evaluate which images are most likely to be target, with the middle of each picture being used as a reference point for all scales. After obtaining the HOG features of each sample, a feature vector x^s is constructed by serially interconnecting features acquired from the retrieved s samples. Furthermore, each x^s has d-dimensional features. Lastly, training samples of different scales in the S layer were passed through to train the scale filter w^s. The (12) has been used to minimize the error of correlation response measured against the desired correlation output $\mathcal{G}$:

$$\varepsilon = \left\|\mathcal{G} - \sum_{s=1}^{d} w^s * x^s\right\|^2 + \lambda \sum_{s=1}^{d} \|w^s\|^2 \tag{12}$$

Where $\mathcal{G}$ is normally the predicted Gaussian response. $\mathcal{G}$, w^s, and x^s have the same dimension and size. The regularization coefficient is denoted byλ. $*$ stands for circular correlation. The (12) can be seen as a linear least squares problem. To solve it, it should be transformed to the Fourier domain. The filter that minimizes (12) is as (13):

$$W^s = \frac{\overline{G}^s}{\sum_{s=1}^{d} \bar{X}^s X^s + \lambda} = \frac{A^s}{B^s} \tag{13}$$

In this situation, the corresponding value discrete Fourier transform is represented by capital letters. G stands for complex conjugation. To solve the problem of redundant or fixed-scale tracking, we have used an estimation scale approach based on swarm-optimization. We have used the ABC algorithm [17], which is one of the most powerful swarm-optimization algorithms. ABC mimics the honeybee swarm's clever foraging behavior. The search mechanism of the ABC algorithm is inspired by the foraging patterns observed in honeybee colonies, utilizing three distinct roles: employed bees, onlooker bees, and scout bees. The process begins with scout bees discovering initial food sources (potential solutions). Employed bees then exploit these sources, sharing information with onlooker bees, which selectively focus on promising solutions based on their fitness (nectar quality). If a food source is exhausted (no further improvement), the employed bee abandons it and becomes a scout to search for a new solution. Crucially, the number of employed bees equals the number of food sources, maintaining a balanced population. The Algorithm 1 iterates through the following phases:

Algorithm 1. ABC algorithm

```
Input: $SN$ (pop. size), MaxCycles, MaxTime
Output: best solution
Initialize $SN$ solutions
while (cycles < MaxCycles and time < MaxTime)
    Employed bees: generate new solutions near existing ones
    Onlooker bees: select solutions proportional to fitness; explore nearby
    Scout bees: replace abandoned solutions with random ones
    Update global best solution
    cycles ← cycles +1
end while
return best solution
```

The ABC method provides a random distribution of food source locations of SN solutions, where SN is the number of onlooker bees or employed bees. Each individual solution takes the structure of a vector defined in d-dimensional space. d denotes the total number of parameters subject to optimization. Afterwards, calculate the value of fitness function for possible solution. In our case, the function (14) of fitness is:

$$y_t^s = F^{-1}\left\{\frac{\sum_{s=1}^{d} \bar{A}_{t-1}^s Z_t^s}{B_{t-1}^s + \lambda}\right\} \tag{14}$$

In the ABC algorithm, a scale pool S is assigned as the x_i vector for each frame of the video. Multiple targets of various scales are chosen from this pool. At frame t, ABC algorithm initializes the vector with random values, where the fitness evaluation is performed using the mathematical expression defined as (14). A_{t-1} and B_{t-1} are the filter coefficients at frame $t-1$. The different steps of our new object tracker are summarized in Algorithm 2.

Algorithm 2. The fundamental process of our algorithm

```
Input: pos (1) the starting location of the object in the initial frame, VGGNet-19 pre-trained models;
Output: Predicted location post; updated correlation filters; updated scale model;
Initiate filter model using (1);
Initiate KF models;
X0 extracted CNN features at initial frame;
for t=2, 3, …. Do
Extracted CNN features at frame t according to pos(t-1).
Equation (5) to calculate the response maps of filter;
Using the calculated pos(t) as recording, predict the target location poskalman(t) using the Kalman Filter (KF);
Extracted CNN features of frame t according to poskalman(t), calculate the response maps regarding (5);
Choose the maximum response map between the two and compute the reliability information using (8) and (5);
if r>Thr then
Use (8) to determine the target location pos(t) at the current frame;
Find the best scale that maximizes (14) based on ABC algorithm
```

```
        Update scale model
        The online training module feeds it with the extracted features X0 and Xt and updates CF;
    else
        Choose pos(t) that has the maximum response between the two.
    end if
end for
```

3. METHOD

3.1. The evaluation of performance

In this section, we will first examine the details of our tracker implementation and the configuration of its parameters. The tracker we suggested was developed in MATLAB 2018b and evaluated on a desktop computer with an Intel i7-4700HQ CPU@2.40 GHz x8 and 8 GB RAM. In order to implement VGGNet-19 [15] convolutional feature extraction, we utilize MatConvNet, a popular MATLAB deep learning framework. Secondly, we will exhibit the tracking performance of our proposed tracker in comparison to various state-of-the-art trackers. We analyze the performance of our system using the OTB-2015 and OTB-2013 datasets, that include attributes-based evaluations, quantitative, and qualitative. To be more precise, the tracker settings are set to the following:

The regularization term parameter is set to λ=0.01, the padding value for the region has been customized to 1.8, the learning rate is η=0.01, and with a value of 0.1 the filter σ kernel width is adjusted. With weights of 1, 0.5, and 0.25, correspondingly, we have utilized the features of conv5-4, conv4-4, and conv3-4 of VGGNet-19. For translation, KF assigns a value of $Q = [25, 10, 10]$, $R = 25$ to the covariances of motion and measured noise. Finally, the control parameters the ABC scale are given in Table 1.

Table 1. ABC algorithm parameters

Parameter	Value
Maximum number of iterations	30
Population size	30
Limit value	20
Acceleration coefficient	1
Dimension	1

3.2. Evaluation criteria

We utilize the OPE criteria [18], [19] to evaluate all trackers on the OTB dataset, which includes two metrics: success rate and Precision. Precision is defined as $\left\|c_t - c_g\right\|$, where c_t and c_g denote the centers of the predicted and ground-truth bounding boxes, respectively. When comparing the performance of different trackers, a 20-pixel distance threshold is typically utilized. The degree of crossover between the two boxes determines the success rate. A tracking instance is deemed successful when the intersection over union (IoU) value reaches or exceeds a predefined threshold, typically set at 0.5. The IoU metric is calculated using the (15):

$$IOU = \frac{area(B_T \cap B_G)}{area(B_T \cup B_G)} \tag{15}$$

The area under the curve (AUC), computed from success plots across varying IoU thresholds, serves as a standard measure of overall tracking quality. Success at the frame level is determined by the overlap ratio (IoU) obtained from comparing the predicted region B_T to the ground truth B_G, where tracking is considered correct if this value surpasses a specified threshold. The success rate is therefore the ratio of correctly tracked frames to the total sequence length.

4. RESULTS AND DISCUSSION

4.1. Ablation experiments

Ablation experiments on the OTB dataset are performed to evaluate the performance of each module described in this article. The DCF tracker is used as a baseline, but features are extracted using a convolution network. For the purpose of the evaluation of different components efficiency, we built three independent trackers by integrating the baseline tracker with the ABC scale and each component: Baseline with ABC scale+RA the baseline and the reliability analysis module are combined, the baseline with ABC scale+KF is produced by combining the baseline with the KF and the last one is baseline with ABC scale+OT signifies that the target information from both the initial frame and the present frame is used to train the updated filter. Moreover, we evaluate the performance of ABC scale technique against the traditional one.

Figures 2 and 3 present a summary of the overall experimental outcomes. Accuracy measurements were carried out experimentally, as seen in the left figure. The tracking accuracy at a distance error threshold of 20 is represented by the number in the legend. The right graph illustrates the overall success rate of each tracker; it is represented by the number in the legend (AUC).

For Figure 2, the legends show the AUC and DP score at 20 pixels for each ablation, based on one-pass evaluation of precision and success plots on the 50 sequences of the OTB-2013 dataset. Figure 3 follows the same evaluation procedure for all 100 sequences of the OTB-2015 dataset, with the legends displaying both the DP score at 20 pixels and the AUC for each ablation. On OTB2013 and OTB2015, Baseline achieves maximum precision plot performance of 86.7% and 85.1%, respectively. Over the other four trackers, OTB2013 has a precision performance increase of 0.9%; 1%; 3.3%; and 7%; whereas OTB2015 has a precision performance improvement of 1.6%; 2.2%; 3.8%; and 5.9%. In the same way, Baseline with ABCscale+All achieves precision scores in the success plots 66.5% and 62.5% on OTB2013 and OTB2015, respectively. OTB2013 shows success increases of 1.1%, 3.9%, 4.7%, and 10%, although OTB2015 shows success gains of 1.5%, 2.0%, 2.3%, and 6.3% in comparison to the other four trackers.

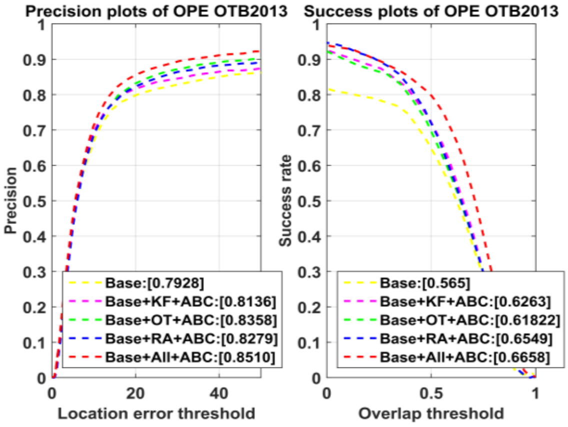


Figure 2. Precision and success plots of the ablation study on the OTB-2013 dataset

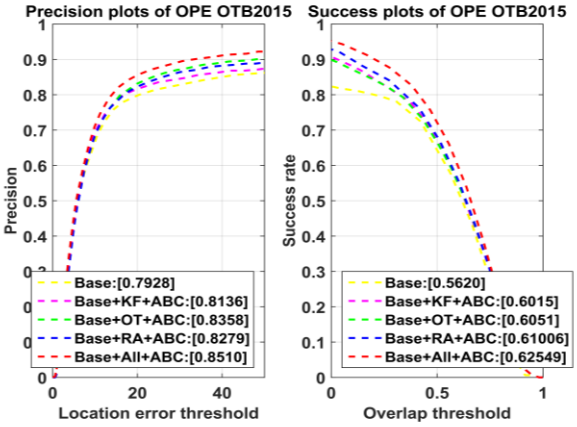


Figure 3. Precision and success plots of the ablation study on the OTB-2015 dataset

KF shows the smallest improvement in benchmark tracker performance relative to the other two modules. The poor precision of the benchmark tracker leads to more measurement error in the KF process, producing mediocre prediction results. Therefore, single KF module enhances the benchmark tracker efficiency slightly compared to the other three modules. Additionally, the OT module makes an important contribution to improving how well the performance tracker works. The module consistently retains the initial frame target information.

According to the results of this study, an effective tracking framework requires accurate information about objects. The RA module has the most impact on improving the efficiency of the baseline tracker. Since it is able to maintain precise object information while avoiding tracking drift and loss in various scenarios (e.g., occlusion and extended sequences).

Figures 4 and 5 show that the paper suggested tracker technique on the OTB-100 data set has comparable benefits: DP rate performance was higher with the tracker we suggested compared to one without the ABC scale, which was 86.7% vs. 85.38% in OTB-2013 and 85.1% vs. 84.53% in OTB-2015. In terms of coverage success rate, the performance was comparable to 66.58% and 63.2%, 62.55% and 62.1% in OTB-2013 and OTB-2015 respectively. In Figure 4, the precision and success plots, annotated with DP scores at 20 pixels and AUC values in their legends, were obtained from experiments conducted on the OTB-2013 benchmark. The analysis was carried out using the standard one-pass evaluation protocol across all 50 sequences for each ablation experiments with and without ABC approach. In Figure 5, the evaluation on OTB-2015 was conducted using one-pass evaluation across 100 sequences, from which both success and precision plots were derived. Legends accompanying these plots display the DP score at 20 pixels along with the AUC values for ablation experiments that either include or exclude the ABC approach.

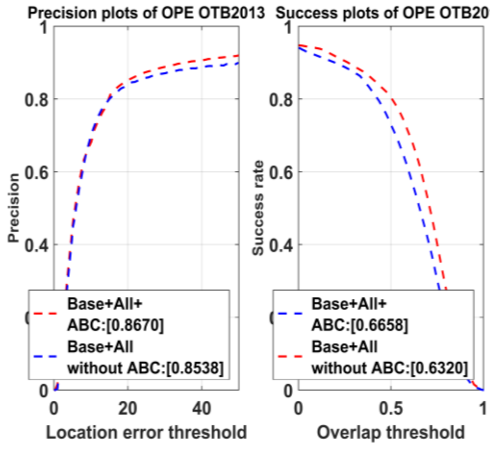


Figure 4. Precision and success plots comparing the tracker with and without the ABC module on the OTB-2013 dataset

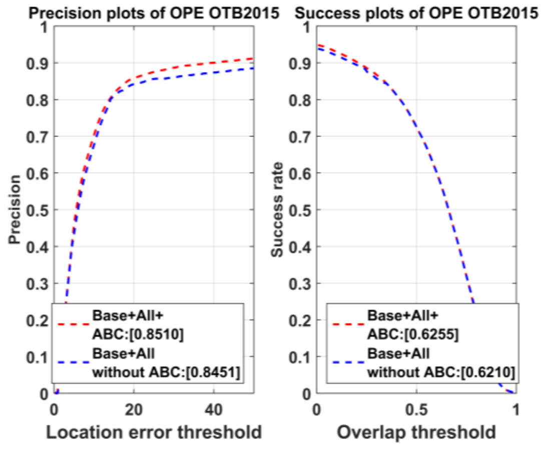


Figure 5. Precision and success plots comparing the tracker with and without the ABC module on the OTB-2015 dataset

The OTB dataset has been meticulously annotated with 11 unique attributes, each of which poses a different difficulty. These features include: occlusion, fast motion, deformation, low resolution, background clutter, scale variation, in-plane rotation, out-of-view, illumination variation, out-of-plane rotation, and motion blur. These attribute-based subsets are crucial for assessing tracker performance and making improvements.

For each tracker, the precision and AUC values are shown in Tables 2 and 3 for the attribute-based subsets. BaselineAll clearly outperforms the challenge across the board, and this holds true for all attribute subsets. As we can see, Baseline+All outperforms all other trackers in every attribute subset. We realized that our proposed framework greatly improved the baseline tracker performance and that the performance increase according to each module was in line with the overall results. In addition, this demonstrates that reliable target information is critical for the tracking process and that KF also supplies critical supplemental information for effective tracking operations.

Table 2. Precision results of OTB100 ablation evaluated on 11 distinct attributes

Attribute	Base+All	Base+RA	Base+OT	Base+KF	Base
Occlusion	0.8020	0.7918	0.7321	0.7748	0.771
Background clutters	0.8452	0.8117	0.8084	0.7985	0.786
Fast motion	0.8050	0.8043	0.7557	0.8032	0.706
Illumination variation	0.8364	0.8052	0.8062	0.7975	0.786
Deformation	0.8164	0.7929	0.8099	0.7459	0.805
In-plane rotation	0.8338	0.8327	0.7720	0.8313	0.789
Scale variation	0.7941	0.7777	0.7365	0.7703	0.754
Low resolution	0.8316	0.8287	0.8099	0.7944	0.737
Motion blur	0.8007	0.7968	0.7566	0.7868	0.736
Out-of-plane rotation	0.8241	0.7808	0.7708	0.7764	0.782
Out-of-view	0.7372	0.6721	0.6214	0.6714	0.667

Table 3. Results of the OTB100 ablation experiment: AUC values across 11 attribute categories

Attribute	Base+All	Base+RA	Base+OT	Base+KF	Base
Occlusion	0.6111	0.6075	0.5997	0.6094	0.539
Background clutters	0.6245	0.6078	0.6030	0.6041	0.583
Fast motion	0.5829	0.5955	0.5936	0.5910	0.539
Illumination variation	0.6140	0.5930	0.5892	0.6088	0.556
Deformation	0.5993	0.6043	0.6032	0.5720	0.531
In-plane rotation	0.6017	0.5852	0.5731	0.5799	0.568
Scale variation	0.5868	0.5745	0.5691	0.5711	0.532
Low resolution	0.5233	0.5188	0.5151	0.5019	0.496
Motion blur	0.6470	0.6333	0.6178	0.6212	0.548
Out-of-plane rotation	0.6078	0.5911	0.5824	0.5918	0.551
Out-of-view	0.6144	0.6092	0.5854	0.5988	0.482

4.2. Comparison with other trackers

To conduct a more thorough analysis and evaluation of the proposed tracker, we perform a comparative study against several representative tracking methods, including DSST [20], IFCT [21], selective part-based correlation filter (SPCF) [22], Zhao *et al.* [23], kernelized correlation filter (KCF) [24], Liu *et al.* [25], Autotrack [26], Ad_SASTCA [27], Shu *et al.* [28], learning adaptive discriminative correlation filters (LADCF)_HC [29], MCMCF [30]. Additionally, we performed experiments using OTB2013 and OTB2015 as well. The proposed tracking method attains superior results on the OTB2013 benchmark, reaching a precision score of 0.8670 and a success rate of 0.6658, as illustrated in Figure 6. In comparison, SPCF reports 0.859/0.628, Liu *et al.* [25] 0.8523/0.630, IFCT 0.835/0.637, Zhao *et al.* [23] 0.825/0.662, Autotrack 0.825/0.619, DSST 0.747/0.557, and KCF 0.740/0.514 in terms of precision and success, respectively. The reported results validate the efficiency of the proposed technique in maintaining accurate and robust tracking across diverse scenarios. Experimental validation on the OTB2015 benchmark confirms that the proposed tracker attains state-of-the-art performance. It secures the highest precision (0.8510) and a competitive success rate (0.6255) among seven evaluated trackers, thereby validating its design objectives for achieving high accuracy and robustness in complex visual tracking environments, as shown in Figure 7.

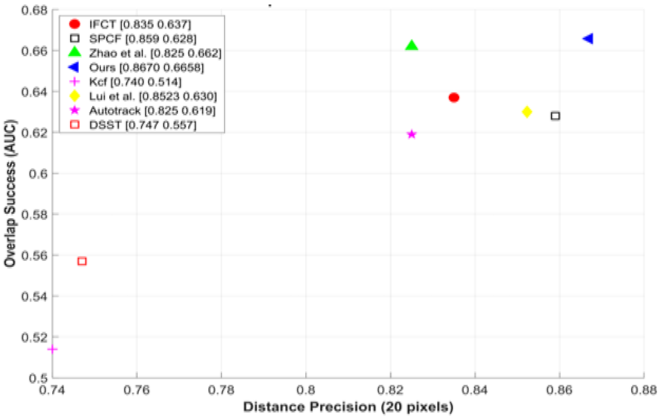


Figure 6. Success (AUC) and precision (20-pixel threshold) comparison of the proposed tracker with seven others on the OTB2013 dataset

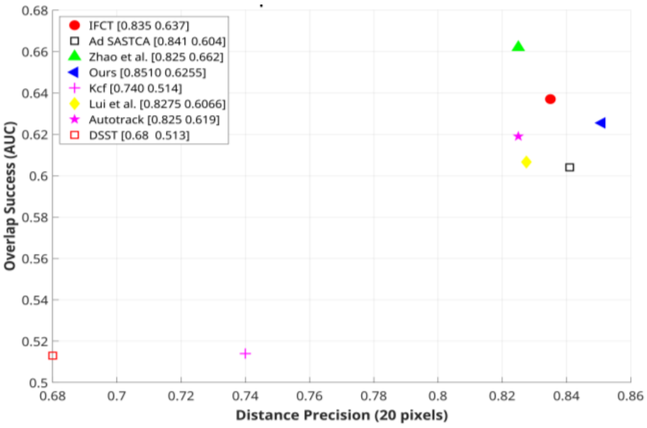


Figure 7. Performance comparison of the proposed tracker with others on the OTB2015 dataset, based on success rates and precision

5. CONCLUSION

This research introduces a robust visual tracking approach consisting of four main components: an examination of reliability module, accurate online training and update module, KF module, and ABC scale method. The reliability analysis module assesses the tracking process, deciding whether to initiate the training update phase to prevent noise introduction. The reliable online training update module merges information from the initial and current frames, preserving trustworthy target information. The KF module simulates inter-frame motion, offering valuable supplemental data. The proposed technique enhances tracking performance in challenging scenarios, addressing appearance changes, tracking drift, and occlusions. The ABC scale approach, based on the ABC optimization algorithm, mitigates scale variation issues in target tracking, enhancing accuracy and resilience. We assessed our proposed tracking framework using the standard benchmark datasets OTB2013 and OTB2015. The results show that our tracking method achieved top performance, securing first place on both OTB2013 and OTB2015 datasets. The tracking results confirm that our approach delivers state-of-the-art performance levels. However, we observed a limitation where our tracker experiences difficulties in maintaining accurate positioning when target objects undergo in-plane rotation. This weakness presents an opportunity for improvement in future development phases.

FUNDING INFORMATION

No funding was provided to support the conduct of this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ramdane Taglout	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Bilal Saoud		✓				✓			✓	✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors affirm that they are not aware of any financial or personal affiliations that could have inappropriately influenced the research and findings presented in this manuscript.

DATA AVAILABILITY

The datasets employed to generate the results of this work are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. S. Hossain *et al.*, “Automatic navigation and self-driving technology in agricultural machinery: a state-of-the-art systematic review,” *IEEE Access*, vol. 13, pp. 94370–94401, 2025, doi: 10.1109/ACCESS.2025.3573324.

[2] K. Rezaee, “Machine learning and facial recognition for down syndrome detection: a comprehensive review,” *Computers in Human Behavior Reports*, vol. 17, 2025, doi: 10.1016/j.chbr.2025.100600.

[3] T.-L. Le, H. N.-Xuan, and S.-A. Kim, “Hand drawing-based daylight analysis using deep learning and augmented reality,” *Results in Engineering*, vol. 22, 2024, doi: 10.1016/j.rineng.2024.102119.

[4] A. Watson and V. W.-O’Connor, “The promise of artificial intelligence in health: portrayals of emerging healthcare technologies,” *Sociology of Health & Illness*, vol. 47, no. 1, 2025, doi: 10.1111/1467-9566.13840.

[5] G. Suresh, P. Reddy, M. Praveen, P. Arun, and K. Jayanth, “Real time room automation using IoT device,” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 14, pp. 422–434, 2023, doi: 10.17762/turcomat.v14i2.13672.

[6] P. Kadam, G. Fang, and J. J. Zou, “Object tracking using computer vision: a review,” *Computers*, vol. 13, no. 6, 2024, doi: 10.3390/computers13060136.

[7] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: a survey,” *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006, doi: 10.1145/1177352.1177355.

[8] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: an experimental survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013, doi: 10.1109/TPAMI.2013.50.

[9] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, “Handcrafted and deep trackers: recent visual object tracking approaches and trends,” *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–44, 2019, doi: 10.1145/3309665.

[10] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, “Robust and fast collaborative tracking with two stage sparse optimization,” in *European Conf. Computer Vision (ECCV)*, 2010, vol. 6314, pp. 624–637, doi: 10.1007/978-3-642-15561-1_44.

[11] Q. Miao, C. Xu, F. Li, W. Zuo, and Z. Meng, “Delayed rectification of discriminative correlation filters for visual tracking,” *The Visual Computer*, vol. 39, no. 6, pp. 1237–1250, 2023, doi: 10.1007/s00371-022-02750-5.

[12] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77, doi: 10.1109/ICINIS.2015.27.

[13] D. Bolme, J. Beveridge, B. Draper, and Y. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550, doi: 10.1109/CVPR.2010.5539960.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, Sep. 2014.

[15] V. N. Boddeti, T. Kanade, and B. V. Kumar, "Correlation filters for object alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2291–2298, doi: 10.1109/CVPR.2013.297.

[16] H. Chen, W. Zhang, and D. Yan, "Robust visual tracking with reliable object information and Kalman filter," *Sensors*, vol. 21, no. 3, 2021, doi: 10.3390/s21030889.

[17] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007, doi: 10.1007/s10898-007-9149-x.

[18] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, and M. Dehghan, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014, doi: 10.1109/TPAMI.2013.230.

[19] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–23, doi: 10.1109/ICCVW.2015.11.

[20] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, 2014, pp. 1-11, doi: 10.5244/C.28.65.

[21] H. Song, "IoT-oriented visual target tracking and supply chain art product design," *Mobile Information Systems*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/3773469.

[22] Z. Lu, G. Yang, D. Liu, J. Yang, Y. Yang, and C. Zhou, "Selective part-based correlation filter tracking algorithm with reinforcement learning," *IET Image Processing*, vol. 16, no. 4, pp. 1208–1226, 2022, doi: 10.1049/ipr2.12405.

[23] J. Zhao, F. Wei, N. Chen, and Z. Zhou, "Spatial and long–short temporal attention correlation filters for visual tracking," *IET Image Processing*, vol. 16, no. 11, pp. 3011–3024, 2022, doi: 10.1049/ipr2.12535.

[24] 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, 2015, doi: 10.1109/TPAMI.2014.2345390.

[25] J. Liu, P. Ye, X. Zhang, and G. Xiao, "Real-time long-term tracking with reliability assessment and object recovery," *IET Image Processing*, vol. 15, no. 4, pp. 918–935, 2021, doi: 10.1049/ipr2.12072.

[26] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Seattle, USA, 2020, pp. 11920–11929, doi: 10.1109/CVPR42600.2020.01194.

[27] Y. Su, J. Liu, F. Xu, X. Zhang, and Y. Zuo, "A novel anti-drift visual object tracking algorithm based on sparse response and adaptive spatial-temporal context-aware," *Remote Sensing*, vol. 13, no. 22, 2021, doi: 10.3390/rs13224672.

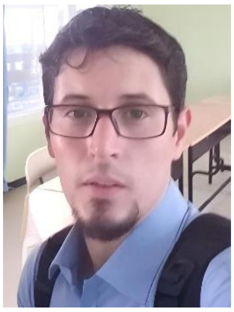
[28] Q. Shu, H. Lai, L. Wang, and Z. Jia, "Multi-feature fusion target re-location tracking based on correlation filters," *IEEE Access*, vol. 9, pp. 28954–28964, 2021, doi: 10.1109/ACCESS.2021.3059642.

[29] M. Liang, X. Wu, Y. Wang, Z. Zhu, B. Cao, and J. Xu, "Multi-model and multi-expert correlation filter for high-speed tracking," *IEEE Access*, vol. 9, pp. 52326–52335, 2021, doi: 10.1109/ACCESS.2021.3069786.

[30] Z. Chen, H. Zheng, X. Zhai, K. Zhang, and H. Xia, "Correlation filter of multiple candidates match for anti-obscure tracking in unmanned aerial vehicle scenario," *Mathematics*, vol. 11, no. 1, 2023, doi: 10.3390/math11010163.

BIOGRAPHIES OF AUTHORS

Ramdane Taglout received his B.Sc. in 2012 from Department of Computer Science and Information Technology, Faculty of New Information Technologies and Communication, Kasdi Merbah University, Ouargla, Algeria. He is currently a Ph.D. student at the Department of Computer Science, University of Bouira. His research interests include visual tracking, computer vision, machine learning, and deep learning. He can be contacted at email: r.taglout@univ-bouira.dz.



Bilal Saoud is an engineer in computer science. He received a Magister's degree and Ph.D. degree in computer science from Bejaia University. He now works as senior lecturer at Bouira University. His main research interests are social networks, application of machine learning, Ad How and wireless sensor networks. He can be contacted at email: bilal340@gmail.com.