

Robust Deep Long-Term Correlation Filter Tracking

Hao Dou¹, Xiwang Guo¹ and Qiangqiang Wu^{2,*}

¹ Discrete Event Systems Laboratory, New Jersey Institute of Technology, Newark, NJ 07102, USA

² Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China

* Correspondence: qiangqwu2-c@my.cityu.edu.hk

How To Cite: Dou, H.; Guo, X.; Wu, Q. Robust Deep Long-Term Correlation Filter Tracking. *Journal of Artificial Intelligence for Automation* 2026, 1(3), 18. <https://doi.org/10.53941/jaia.2026.100018>

Received: 15 May 2026

Revised: 19 September 2026

Accepted: 13 October 2026

Published: 23 October 2026

Abstract: Long-term object tracking is an essential task for developing robust vision ability for intelligent robots and vehicles. To cope with various challenges in the dynamic driving environment, existing methods usually maintain a target appearance model and update it in an online manner. However, the errors may accumulate during the updating process when the target suffers from the problems caused by such issues as long-term occlusion and out-of-view, thus resulting in tracking failures. This work proposes a Robust Deep Long-term Tracking (RDLT) method, which can alleviate the impact of erroneous updates by incorporating an offline learned deep Siamese network into a unified correlation filter tracking framework. Specifically, it learns a deep correlation filter model that models target temporal variations to locate the target. To avoid tracking failures caused by the challenges like long-term occlusion, a pre-trained deep Siamese network is activated to search for the target on an enlarged searching region when the correlation filter outputs low response values, by which several target candidate windows can be obtained. To correctly select the optimal target window, this work further proposes a novel multi-expert evaluation schema. It evaluates the target candidate windows by using a pair of “experts”, i.e., Correlation filter-based expert (C-expert) and Deep similarity-based expert (D-expert). Both experts jointly evaluate all the target candidate windows and select the optimal one as the final tracking result, thereby avoiding the correlation filter being greatly affected by corrupt samples. Extensive experiments on three popular benchmarks including OTB-2013, OTB-2015 and UAV20L, show that the proposed RDLT method performs favorably against the other state-of-the-art methods.

Keywords: long-term object tracking; correlation filters; multi-expert evaluation

1. Introduction

Visual object tracking is one of the fundamental computer vision tasks and it has been the essential part of the vision system utilized by the current intelligent robots and vehicles [1–4], due to its accurate localization ability of surrounding objects, e.g., vehicles, pedestrian and road. Given the initial state of a target at the first frame, visual object tracking is to estimate the states of the target at subsequent frames [5]. Generally speaking, considering the length of videos that object tracking methods can handle with, there are roughly two types of object tracking methods: (1) short-term object tracking methods (e.g., [6,7]) and (2) long-term object tracking methods (e.g., [8,9]). Compared with the former, the latter is much more challenging as objects may suffer from long-term occlusion and out-of-view, which may cause erroneous updates and thus lead to the drifting of the target models. In this paper, we focus on the task of long-term object tracking.

Object tracking methods usually suffer from the problem of significant appearance variations caused by different factors such as rotation, fast motion, motion blur and illumination change. Recently proposed tracking methods [7,10–12] address this problem by constructing some effective appearance models in an online manner and

utilizing the tracking results to update the appearance models during the tracking process. Although these methods have shown high effectiveness in handling with target appearance variations, they may easily drift when targets undergo heavy occlusion and out-of-view during long-term tracking. The main reason is that the tracking errors usually accumulate during the updating of the appearance model especially when targets undergo heavy occlusion and out-of-view, which directly leads to the drifting of the tracking methods. Furthermore, these tracking methods are hard to recover from tracking failures.

To recover from tracking failures for long video sequences, existing long-term tracking methods [8,9] mainly employ online detectors to re-detect targets when tracking failures occur. The key idea behind these methods is that the tracking problem is treated as a classification problem. The tracking performance can be significantly improved by training an online classifier to re-detect the target. For example, the Tracking-Learning-Detection (TLD) method [8] decomposes the task of object tracking into three components, i.e., tracking, learning and detection. In order to reduce the tracking failures, an object detector is used to re-initialize the base tracker employed in TLD. Meanwhile, the tracking results are used to train the detector in an online manner. The experimental results on the tracking dataset [13] show that TLD is well-suited for the long-term object tracking task. In contrast, the long-term correlation tracking (LCT) method [14] trains a standard correlation filter model for translation estimation and uses another correlation filter for scale estimation. Moreover, LCT trains a random fern classifier in an online manner for target re-detection. Although these tracking methods have shown great effectiveness in long-term object tracking, they may fail to re-detect the target when the target undergoes long-term occlusion, which is mainly caused by the performance degradation of the online detectors employed by the methods via erroneous updates. Additionally, the features used in the above tracking methods are hand-crafted, and they are not effective enough to distinguish the target from the complex background when the target undergoes significant target appearance variations due to illumination change, rotation, motion blur, etc.

In recent years, Convolutional Neural Networks (CNNs) have been applied to many computer vision tasks, including object tracking. For example, SINT [15], DiMP [16], and SiamFC [17] are based on the deep similarity learning technique, which uses a similarity function learned by a deep Siamese network to match the target candidate windows at a new frame with the target model initialized at the first frame. Different from the traditional tracking methods, SINT and SiamFC use a large number of training data to train a robust model in an offline manner to adapt to online object tracking. Note that these CNN-based tracking methods do not perform model updating during the tracking process, but they can still achieve favorable performance. The main reason is that these methods use a data-driven strategy, which uses large-scale training data to learn robust feature representations for object tracking. Although these methods do not suffer from the problem of erroneous updates due to their offline training procedure, they still have some limitations. The main limitation is that they are sensitive to significant appearance variations, since they heavily rely on the initial state of the target at the first frame. Therefore, they may easily drift when the target appearance significantly changes or when some similar objects appear in the neighboring region.

In this paper, we propose a robust deep long-term tracking (namely RDLT) method that can effectively alleviate the impact of erroneous updates, which is mainly caused by long-term occlusion and out-of-view. The proposed RDLT method incorporates a pre-trained deep Siamese network into a unified correlation filter tracking framework, which takes advantage of both the online learned correlation filter model and the offline trained Siamese network. Note that the correlation filters can adapt to appearance variations by maintaining an appearance model in an online manner, but it has the problem of erroneous updates when targets undergo long-term occlusion or out-of-view. Meanwhile, the pre-trained deep Siamese network has complementary traits: It does not suffer from the problem of erroneous updates because it does not have online updating, but it may easily drift when the target appearance significantly changes. Therefore, the correlation filter and the pre-trained deep Siamese network are complementary to each other. By taking advantage of the pre-trained Siamese network, the erroneous updating problem of the correlation filter can be effectively alleviated. Thus, our RDLT method can not only encode target appearance variations in an online manner, but also effectively alleviate the impact of erroneous updates.

The overall procedure of RDLT is shown in Figure 1. As we can see, the target undergoes full occlusion from the 348-th frame to the 390-th frame in the lemming sequence. At the 390-th frame, the target reappears. For this challenging case, the correlation filter fails to track the target due to long-time erroneous updates, where the response value of the target region is lower than that of the background region. The proposed RDLT then activates the deep Siamese network to perform re-detection on an enlarged searching region, by which several target candidate windows are obtained. To effectively select the optimal target window, the proposed multi-expert evaluation schema uses both the D-expert and the C-expert to jointly evaluate all the target candidate windows, by which the correct target window is obtained at the #390 frame. By updating the correlation filter model in RDLT with the correct target window, it successfully responds to the target region at subsequent frames (e.g., the #391 frame).

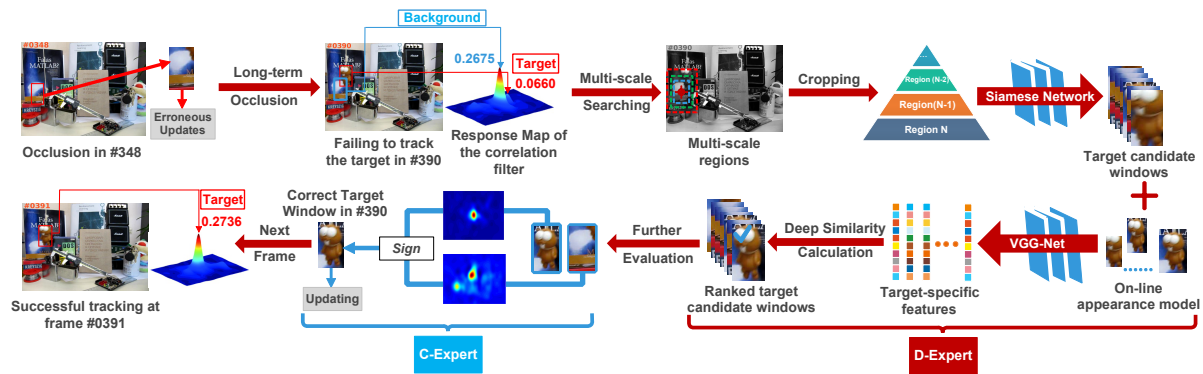


Figure 1. An illustration of how RDLT handles with erroneous updates of the correlation filter (i.e., the base tracker in RDLT) caused by long-term occlusion and out-of-view. The correlation filter fails to track the target at the 390-th frame due to erroneous updates caused by long-term occlusion. By performing re-detection on multi-scale regions and selecting the optimal target window with a multi-expert evaluation schema, RDLT successfully obtains the correct target window at the 390-th frame and tracks the target well at subsequent frames.

We intend to make the following contributions to the field of long-term object tracking:

- (1) We propose a deep long-term tracking method by embedding a pre-trained deep Siamese network into a unified correlation filter tracking framework, which takes advantage of both the correlation filter model and the deep Siamese network. The proposed method can effectively alleviate the problems caused by long-term occlusion and out-of-view, which usually occur in the long-term tracking tasks; and
- (2) We develop a novel multi-expert evaluation schema consisting of two experts (i.e., the deep similarity-based D-expert and the correlation filter-based C-expert). Both experts are jointly used to select the optimal target window from target candidate windows, by which it avoids that the correlation filter model is greatly affected by corrupt samples. Thus, the proposed RDLT method is effective to handle with erroneous updates.

Extensive experiments on three popular benchmarks (i.e., OTB-2013 [13], OTB-2015 [5] and UAV20L [18]) show that RDLT outperforms several state-of-the-art trackers in terms of both accuracy and robustness. Further evaluation on some long video sequences demonstrates that RDLT copes well with the challenges of the long-term tracking.

2. Related Work

In this section, we introduce the approaches closely related to our work. We review correlation filter-based trackers, which form the basis of our online appearance model, and then summarize the recent progress on Transformer-based trackers, which represent the dominant trend in modern visual tracking.

2.1. Correlation Filter Tracking

Correlation filter (CF) based trackers have attracted considerable attention due to their high computational efficiency. Bolme et al. [19] first propose the Minimum Output Sum of Squared Error (MOSSE) method, which uses only grayscale images to train the regression model and reaches hundreds of frames per second. Henriques et al. extend this idea by introducing the CSK [20] and the kernelized correlation filter KCF [7], which lift the input from a single channel to multiple channels and exploit the kernel trick to achieve a favorable trade-off between accuracy and speed. Since hand-crafted features are not always discriminative enough to distinguish a target from cluttered backgrounds, several works replace them with deep convolutional features. Both CF2 [10] and Deep-SRDCF [12] use the multi-layer features extracted by the VGGNet [21], outperforming hand-crafted feature based CF trackers [22] by large margins on [13]. HDT [11] further hedges multiple correlation filters trained on different convolutional layers via an improved Hedge algorithm [23]. To explicitly cope with long-term tracking, TLD [8] couples a base tracker with an online detector that re-initializes the tracker when failures occur, while LCT [14] pairs a translation correlation filter with a scale filter and a random fern re-detector. Although these methods can recover from short-term failures, the errors arising from the erroneous online updates of their base trackers and detectors still accumulate when the target undergoes long-term occlusion and out-of-view, which directly causes model drifting. In addition, deep similarity-based trackers such as SINT [15] and SiamFC [17] avoid the model-drift problem by matching candidates against a fixed template learned offline, but they are sensitive to large appearance variations because they rely heavily on the initial target state. These observations motivate us to combine deep correlation filters with a pre-trained deep Siamese network in a unified long-term tracking framework.

2.2. Siamese Tracking

Siamese-network-based trackers have become one of the dominant paradigms for visual object tracking, mainly because they cast tracking as a learnable similarity matching problem and can be trained on large-scale video data in a fully offline manner. Building upon SiamFC [17], which formulates tracking as cross-correlation between a target template and a search region, a number of recent works further improve both the localization accuracy and the robustness of Siamese trackers. SiamCAR [24] proposes a fully convolutional classification-and-regression Siamese framework that directly predicts the target bounding box at every spatial location without using pre-defined anchors, which simplifies the tracking pipeline and effectively handles target scale variations. SiamBAN [25] introduces a box-adaptive anchor-free Siamese network in which the bounding-box regression is performed in an end-to-end manner, further improving the tracking accuracy on challenging benchmarks. To extend Siamese tracking beyond standard visual benchmarks, SANVOT [3] develops a Siamese adaptive network for accurate and robust visual object tracking on quadrupedal robots, which demonstrates the practical value of Siamese trackers for real-world robotic platforms. Although these Siamese trackers achieve favorable performance on short-term benchmarks, they typically do not update the target model in an online manner and therefore remain sensitive to long-term appearance variations and severe occlusion. In this work, we exploit a pre-trained deep Siamese network as the re-detection module within a correlation filter framework, so as to take advantage of both the offline-learned discriminative similarity and the online-updated target appearance model.

2.3. Transformer Tracking

Inspired by the great success of Transformers [26] in natural language processing and the introduction of the Vision Transformer [27], recent works have explored attention mechanisms to build powerful visual trackers. Chen et al. [28] propose TransT, which replaces the cross-correlation operation in Siamese trackers with an ego-context augment and cross-feature augment module based on attention, leading to substantial improvements over correlation-based counterparts. Yan et al. [29] introduce STARK, an encoder-decoder Transformer that jointly models spatial and temporal information by maintaining a dynamically updated template, which alleviates the appearance variations encountered during long-term tracking. Wang et al. [30] apply Transformer-based encoder-decoder modules on top of Siamese and discriminative trackers to exploit rich temporal context across frames. More recent works further unify feature extraction and relation modeling within a single backbone. MixFormer [31] interleaves the template and search regions through iterative mixed attention, while OTrack [32] performs joint feature learning and relation modeling in a one-stream Vision Transformer with candidate elimination for efficiency. Other notable advances include ToMP [33], which uses a Transformer to predict the weights of the target localization model on the fly, SwinTrack [34], a fully-attentional baseline built upon the Swin Transformer, and SeqTrack [35], which casts visual tracking as a sequence generation task.

Beyond architectural design, recent works extend Transformer-based tracking along several complementary directions, including video-oriented self-supervised pre-training [36], weakly-supervised learning from mask-to-point [37] or single-point [38] annotations, fully unsupervised target mining [39], simplified frameworks that unify tracking and video object segmentation [40], lightweight deep–shallow feature aggregation for efficient tracking [41], and graph-based meta-adaptation to new targets [42]. A parallel line of research further casts tracking as a coordinate or token sequence prediction problem, e.g., ARTrack [43] and its successor ARTrackV2 [44] estimate the target trajectory in an autoregressive manner conditioned on previous states, ODTrack [45] learns dense temporal tokens online to propagate target information across frames, and AQATrack [46] introduces autoregressive queries with spatio-temporal Transformers to adapt the tracker on the fly. The growing emphasis on segmentation-based evaluation is further reflected in the recent VOTS2023 [47] and VOTS2024 [48] challenges, while studies on temporal unlearnable examples [49] additionally raise emerging concerns about unauthorized tracking of personal videos. Although these Transformer-based trackers and their extensions achieve remarkable accuracy on short-term benchmarks, most of them rely on a fixed or slowly updated template and do not explicitly address the long-term failure cases caused by heavy occlusion and out-of-view, which are the focus of our work.

3. Deep Long-Term Object Tracking

The proposed RDLT method consists of three main components: (1) A base tracker that effectively handles target temporal appearance variations in an online manner; (2) A target detector that is trained in an offline manner; and (3) a multi-expert evaluation schema, which is used to evaluate all target candidate windows obtained by both the base tracker and the detector. We use the deep correlation filter as the base tracker, and use the pre-trained deep Siamese network as the target detector. In this section, we detail the three components and the implementation details of the proposed RDLT method.

3.1. Deep Correlation Filter for Tracking

Due to the effectiveness of deep correlation filters in encoding the temporal context information between consecutive frames, we train a deep correlation filter as in [10] as our base tracker. Let $\mathbf{x}^l$ denote the feature maps extracted from the l -th layer of the VGGNet [21], $\mathbf{x}^l$ can be expressed as $\mathbf{x}^l = \{\mathbf{x}_{m,n,d}^l | (m, n, d) \in \{1, 2, \dots, M\} \times \{1, 2, \dots, N\} \times \{1, 2, \dots, D\}\}$, where M, N and D denote the width, height and the number of channels, respectively. The deep correlation filter learns a correlation filter model by solving a Ridge Regression problem:

$$\hat{\mathbf{w}}^l = \min_{\mathbf{w}^l} \sum_{m=1}^M \sum_{n=1}^N \left[\sum_{d=1}^D \mathbf{w}_{m,n,d}^l \mathbf{x}_{m,n,d}^l - \mathbf{y}(m, n) \right]^2 + \lambda \|\mathbf{w}^l\|_2^2, \quad (1)$$

where $\hat{\mathbf{w}}^l$ is the correlation filter model trained with the feature maps on the l -th layer of the VGGNet, and it has the same size as $\mathbf{x}^l$. λ is a regularization parameter and $\mathbf{y}(m, n)$ yields a continuous label by a Gaussian function. That is, $\mathbf{y}(m, n) = e^{-\frac{(m-M/2)^2 + (n-N/2)^2}{2\sigma^2}}$, where σ is the linear kernel width. The minimization problem in (1) can be solved by using the fast Fourier transform algorithm [10], and the learned correlation filter model in the frequency domain on the d -th channel can be written as:

$$\hat{\mathbf{W}}_d^l = \frac{\mathbf{Y} \odot \bar{\mathbf{X}}_d^l}{\sum_{i=1}^D \mathbf{X}_i^l \odot \bar{\mathbf{X}}_i^l + \lambda}, \quad (2)$$

where the capital bold letters denote the discrete Fourier transform of the corresponding lowercase letters, $\bar{\mathbf{X}}_d^l$ denotes the complex conjugation of $\mathbf{X}_d^l$ and $\odot$ denotes the element-wise product. For a test image, let $\mathbf{z}^l$ denote the feature maps on the l -th layer and it has the same size as $\mathbf{x}^l$, where $\mathbf{z}^l = \{\mathbf{z}_{m,n,d}^l | (m, n, d) \in \{1, 2, \dots, M\} \times \{1, 2, \dots, N\} \times \{1, 2, \dots, D\}\}$. Since the feature maps from different layers have different sizes, the bilinear interpolation method is employed to resize different feature maps into the same size. Then the response map generated based on the feature maps on the l -th layer can be computed by:

$$f_l = \mathcal{F}^{-1} \left(\sum_{d=1}^D \hat{\mathbf{W}}_d^l \odot \bar{\mathbf{Z}}_d^l \right), \quad (3)$$

where f_l is the response map on the l -th layer, the operator $\mathcal{F}^{-1}$ denotes the inverse Fourier transform and $\bar{\mathbf{Z}}_d^l$ is the discrete Fourier transform of $\mathbf{z}_d^l$. By using (3), a set of response maps $\{f_l\}_{l=1}^L$ on different layers is obtained, where L is the number of layers we use in the VGGNet. The final optimal position of the target obtained by the deep correlation filter is formulated as:

$$(\hat{m}, \hat{n}) = \arg \max_{m, n} \sum_{l=1}^L \gamma_l f_l(m, n), \quad (4)$$

where $(\hat{m}, \hat{n})$ is the estimated target position in the test image and γ_l is the weight of the response map in the l -th layer of the VGGNet. The deep correlation filter model is then updated by the following equations:

$$\mathbf{A}_{d,t}^l = (1 - \eta) \mathbf{A}_{d,t-1}^l + \eta \mathbf{Y} \odot \bar{\mathbf{X}}_{d,t}^l, \quad (5)$$

$$\mathbf{B}_{d,t}^l = (1 - \eta) \mathbf{B}_{d,t-1}^l + \eta \sum_{i=1}^D \mathbf{X}_{i,t}^l \odot \bar{\mathbf{X}}_t^i, \quad (6)$$

where η is a learning rate and $\bar{\mathbf{X}}_{d,t}^l$ indicates $\bar{\mathbf{X}}_d^l$ at the t -th frame. $\mathbf{A}_{d,t}^l$ and $\mathbf{B}_{d,t}^l$ respectively indicate the numerator and denominator in Equation (2) at the t -th frame.

Usually, response values indicate the tracking failure or success of the deep correlation filter. An example illustrating the response values obtained by the trained deep correlation filter is shown in Figure 2. The deep correlation filter may fail when the target undergoes complicated cases such as deformation, motion blur and heavy occlusion, where the corresponding response values obtained by the deep correlation filter are low. Therefore, instead of employing the target detector at each frame, the proposed RDLT method only activates the detector to search for the target in the cases where the base tracker has low response values. In the next subsection, we will give the details of the target detector employed by the proposed method.

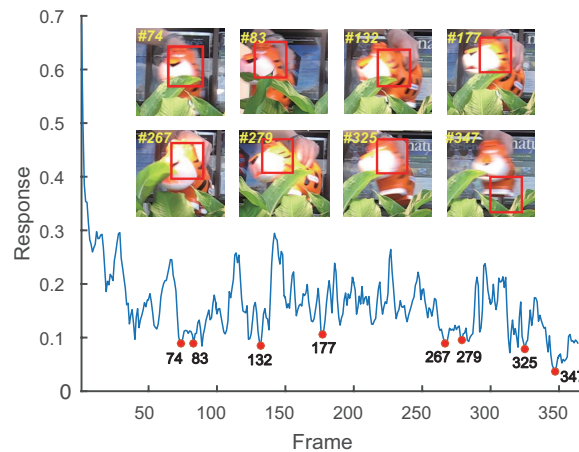


Figure 2. An illustration of the response values obtained only by the deep correlation filter on the Tiger2 sequence. The blue line shows the varying response values during the tracking process. The red dots indicate the low response values at the corresponding frames (i.e., the black numbers). We show the qualitative results at the 74-th, 83-th, 132-th, 177-th, 267-th, 279-th, 325-th and 347-th frames, where the target suffers from deformation, motion blur, occlusion, etc. At these frames, the response values are low, especially at the 347-th frame, the deep correlation filter loses the target.

3.2. Deep Siamese Network for Re-Detection

The deep Siamese network matches the target candidate windows at a new frame with the target model. Since the deep Siamese network can be trained in an offline manner and it has no risks of online erroneous updating, we employ the deep Siamese network in [17] as our target detector to alleviate the erroneous updating problem of the base tracker. Furthermore, to obtain more reliable target candidate windows, we employ the deep Siamese network to search for the target in multiple searching regions.

Given the initial target model, the detector in the proposed method re-detects the target region, which has the maximum similarity score, from the target candidates. The similarity between two image regions can be effectively measured by the deep Siamese network. The Siamese network is trained with large-scale training data in an offline manner and then it measures the similarity between the target candidates and the target model.

The deep Siamese network used in our method contains two branches of CNNs. In the searching process, the two branches are respectively input with a searching region p and a target region o . Thanks to the cross-correlation layer employed by the Siamese network, a scalar-valued similarity score map $S(p, o)$, which calculates the similarity between o and each translated sub-region within the search region p , is obtained in only one pass. Then the location corresponding to the highest score in $S(p, o)$ can be determined and the corresponding sub-region is considered as the most similar region to the target region o . Note that in the practical tracking process for videos with fast motion, the target may undergo fast motion and move out of the searching region. In such a case, to ensure that the target region is covered by the searching regions, we extend the deep Siamese network with Q searching scales. The set of the searching regions with Q different searching scales is defined as $I = \{i_q\}_{q=1}^Q$. For each searching region $i_q \in I$, the center position of the sub-region, which is the most similar region to the target region z in the t -th scale of the searching region, can be calculated as:

$$(\hat{m}_q, \hat{n}_q) = \arg \max_{m, n} (\phi_{i_q, o}(m, n)), \quad (7)$$

where $\phi_{i_q, o}(m, n)$ returns the similarity score at the position (m, n) of the similarity score map $S(i_q, o)$. Let s_q^* denote the similarity score between the target region o and the optimal sub-region centered at $(\hat{m}_q, \hat{n}_q)$ at the q -th scale of the searching region. By searching for the target in all the searching regions in I , we can obtain the center positions of the top- K ranked target candidate windows, which are sorted according to similarity scores $\{s_q^*\}_{q=1}^Q$. Let the center position set of the top- K ranked candidate windows be $U = \{\hat{m}_k, \hat{n}_k\}_{k=1}^K$. Briefly, we use U to represent the set of all the target candidate center positions estimated by both the deep correlation filter and the detector, where we set $(\hat{m}, \hat{n})$, which is the estimated optimal target position by the deep correlation filter, to $(\hat{m}_0, \hat{n}_0)$ and U is then rewritten as $U = \{\hat{m}_k, \hat{n}_k\}_{k=0}^K$.

Additionally, although the target detector can effectively re-detect the target and generate reliable target candidate windows in many complicated cases, it easily fails when the appearance of the target significantly changes and in the meanwhile a spurious object (i.e., a distractor), which is similar to the initial target, appears. Therefore, to

avoid taking a wrong target candidate window as the final tracking result, we propose a discriminative multi-expert evaluation schema, which will be introduced in more detail in the next subsection, to further evaluate the target candidate windows.

3.3. Multi-Expert Evaluation Schema

To effectively evaluate the target candidate windows centered at the target candidate center positions in U and select the optimal target window at each frame, we propose a novel multi-expert evaluation schema, which is composed of a pair of “experts”, i.e., a deep similarity based expert (D-expert) and a correlation filter based expert (C-expert).

3.3.1. D-Expert

D-expert calculates the accumulated similarity distance to each target candidate window centered at the target candidate center positions in U based on an online target appearance model. The online target appearance model consists of target image patches sampled from previous frames, and it includes three kinds of target image patches: (i) the initial target image patch at the first frame; (ii) the latest target image patch that is sampled from the previous frame; (iii) the high-quality target image patches, which are sampled at every ten frames, if $\tau_s > \max(\sum_{i=1}^L \gamma_i f_i)$ is satisfied, where f_i is obtained by (3), L and γ_i are respectively the number of layers and the weight of the i -th response map in (4), and τ_s is a constant value that indicates the tracking failure of the deep correlation filter (e.g., as illustrated in Figure 2);

The D-expert uses the features extracted from the last fully-connected layer of the VGGNet [21], and the features encode the semantic information of objects so that they are robust to significant appearance variations. We extract the semantic features from the target image patches in the online target appearance model. Let $V_0 = \{v_k^0\}_{k=1}^{|V_0|}$, $V_1 = \{v_k^1\}_{k=1}^{|V_1|}$ and $V_2 = \{v_k^2\}_{k=1}^{|V_2|}$ respectively denote the feature sets belonging to the three kinds of target image patches in the online appearance model, where $|\cdot|$ denotes the cardinality of the set. Let the feature set V represent all the three feature sets (i.e., V_0 , V_1 and V_2), where $V = \{v_k\}_{k=1}^{|V_0|+|V_1|+|V_2|}$. Since the target scale χ_{t-1} (i.e., the width and height of the target window) at the $(t-1)$ -th frame is close to the target scale at the t -th frame, we crop the target candidate windows (i.e., those centered at the target candidate center positions in U) at the t -th frame with the scale χ_{t-1} . The updating process of the target scale at the t -th frame is given in Sec. III-D. By inputting the target candidate windows into the VGGNet, the feature set $E = \{e_k\}_{k=0}^K$ (i.e., the outputs of the last fully-connected layer in the VGGNet) is obtained. For each $e_k \in E$, the D-expert calculates its accumulated similarity distance to the online target appearance model based on V as follows:

$$d(e_k) = \sum_{i=1}^{|V_0|+|V_1|+|V_2|} \sqrt{(v_i - e_k)(v_i - e_k)^T}, \quad (8)$$

and then the index of the optimal target candidate window estimated by the target detector of RDLT is obtained by:

$$\hat{k} = \arg \min_k d(e_k), \quad (9)$$

the D-expert further evaluates the two accumulated similarity distance values of $d(e_{\hat{k}})$ and $d(e_0)$ by the following equation:

$$r_D = \text{sign}(d(e_0) - d(e_{\hat{k}})), \quad (10)$$

where e_0 is the feature vector of the target candidate window estimated by the deep correlation filter, r_D is the evaluation value obtained by the D-expert, and $\text{sign}(\cdot)$ is the sign function. Thus if the optimal target window obtained by the target detector in RDLT is more reliable than the target window estimated by the deep correlation filter, the evaluation value r_D is set to 1, and the optimal target candidate window estimated by the detector and the target candidate window estimated by the base tracker will be further evaluated by the C-expert. Otherwise, the target window estimated by the base tracker will be directly taken as the final tracking result.

3.3.2. C-Expert

The C-expert further evaluates two optimal target candidate windows, i.e., the optimal target window obtained by the detector and the target window estimated by the base tracker. The C-expert consists of two deep correlation filters, where the first one is our base tracker, which is updated during the whole tracking process, and the second one is only trained at the first frame. Let $\tilde{C}_t$ denote the deep correlation filter trained at the whole t frames and $\tilde{C}_1$

denote the deep correlation filter trained at the first frame, respectively. It is important to note that $\tilde{C}_t$ takes the target appearance variations into consideration, while $\tilde{C}_1$ only keeps the original target model at the first frame and it has no risks of erroneous updating, which may help $\tilde{C}_t$ recover from the tracking failures caused by the erroneous updates. In addition, when the appearance of the target significantly changes, $\tilde{C}_1$ is still able to work. This is because that $\tilde{C}_1$ uses the convolutional features extracted from the later layers of the VGGNet, which encode more semantic information and are robust to appearance variations. Let $R_t(m, n)$ and $R_1(m, n)$ be the response values obtained by $\tilde{C}_t$ and $\tilde{C}_1$ on the target window centered at the position (m, n) in an image, respectively. $R_t(m, n)$ and $R_1(m, n)$ can be formulated as:

$$R_t(m, n) = \sum_{l=1}^L \gamma_l f_l^t(m, n), \quad (11)$$

$$R_1(m, n) = \sum_{l=1}^L \gamma_l f_l^1(m, n), \quad (12)$$

where $f_l^t(\cdot)$ and $f_l^1(\cdot)$ are the response maps obtained by $\tilde{C}_t$ and $\tilde{C}_1$ in the l -th layer of the VGGNet, respectively. The evaluation of the response values at the two optimal target candidate center positions $(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}})$ and $(\hat{m}_0, \hat{n}_0)$ by the C-expert can be written as:

$$r_C = (\text{sign}(R_t(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}}) - R_t(\hat{m}_0, \hat{n}_0)) \vee (\text{sign}(R_1(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}}) - R_1(\hat{m}_0, \hat{n}_0))), \quad (13)$$

where r_C is the evaluation value of the C-expert for the optimal target window obtained by the target detector and the target window estimated by the base tracker. If $r_C = 1$, the C-expert selects $(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}})$ obtained by the detector as the final tracking result. Otherwise, the C-expert selects $(\hat{m}_0, \hat{n}_0)$ obtained by the base tracker as the final result. More specifically, when $R_t(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}}) > R_t(\hat{m}_0, \hat{n}_0)$, it means that the target window obtained by the target detector is more reliable than that estimated by the deep correlation filter. This is because that the former one takes more target appearance variations into consideration. In another case, i.e., when $R_1(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}}) > R_1(\hat{m}_0, \hat{n}_0)$, there is a high probability that the deep correlation filter is erroneously updated. In this case, the optimal target window obtained by the target detector is a better choice.

The overall procedure of the proposed multi-expert evaluation schema is shown in Algorithm 1. When the optimal target window obtained by the target detector is accepted as the final result for a frame with a high response value (i.e., $(\sum_{l=1}^L \gamma_l f_l^t(\hat{m}, \hat{n})) > \tau_s$), the final selected tracking result is used to update the base tracker with the adjusted learning rate η' (i.e., $\eta' = \alpha \times \eta$, where α serves as a learning rate) using (5) and (6), which aims to alleviate the impact of the erroneous updates.

Algorithm 1: Multi-expert Evaluation Schema

Input: The target candidate center position set U_t at the t -th frame;
Output: The optimal target position (m_t, n_t) and the evaluation value r_C by the C-expert.

- 1 Initialize: $r_C \leftarrow 0$ and $accept \leftarrow \text{False}$;
- 2 Extract the features of the target candidate windows centered at the positions in U_t , and obtain the feature set E_t at the t -th frame;
- 3 Calculate the accumulated similarity distance for each $e_k \in E_t$ using (8) and obtain the index $\hat{k}$ of the optimal target candidate window;
- 4 Compute the evaluation value r_D using (10);
- 5 **if** $r_D = 1$ **then** // C-expert
 - 6 Use the C-expert to evaluate target candidate windows centered at $(\hat{m}_0, \hat{n}_0)$ and $(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}})$ using (11) and (12), respectively;
 - 7 Compute the evaluation value r_C using (13);
 - 8 **if** $r_C = 1$ **then** // Accept
 - 9 $accept \leftarrow \text{True}$;
- 10 **if** $accept = \text{True}$ **then**
 - 11 Select $(\hat{m}_{\hat{k}}, \hat{n}_{\hat{k}})$ obtained by the detector as the final result (m_t, n_t) at the t -th frame;
- 12 **else**
 - 13 Select $(\hat{m}_0, \hat{n}_0)$ estimated by the base tracker as the final result (m_t, n_t) at the t -th frame;
- 14 **return** $(m_t, n_t), r_C$;

3.4. Implementation

In this subsection, we discuss the implementation details of the proposed RDLT method.

3.4.1. Tracking Procedure

We present the main tracking procedure of RDLT in Algorithm 2. In the practical long-term tracking process, RDLT often suffers from the challenges of deformation, long-term occlusion, out-of-view, etc. Usually, we can effectively deal with these problems by using the response values of the base tracker and implementing the step of re-detection when the deep correlation filter yields low response values (i.e., $R_1(\hat{m}, \hat{n}) < \tau_s$). However, the base tracker may fail to track the target with high response values. To avoid these tracking failures, RDLT performs an additional step of re-detection every ten frames when $\tau_s < R_1(\hat{m}, \hat{n})$, where τ_s is the threshold of the low response value. Note that the base tracker employed in RDLT considers that the center position of the target only varies in translation and no scale variation occurs. When obtaining the target's center position at a new frame, we follow Danelljan et al. [50] and learn a multi-scale template correlation filter to estimate the scale of the target. This correlation filter uses HOG features for computational efficiency, and it has been widely used in many correlation filter based tracking methods for scale estimation.

Algorithm 2: RDLT

Input: The target center position (m_1, n_1) at the first frame, the deep correlation filter model trained at the first frame, the pre-trained deep Siamese network model and the pre-trained VGGNet-19 model.

Output: The estimated target center position (m_t, n_t) at the t -th frame.

```

1 for  $t = 2, 3, \dots$  do
2   Estimate the initial target position  $(\hat{m}, \hat{n})$  at the  $t$ -th frame by the deep correlation filter  $\tilde{C}_t$  using (4);
3   Initialize:  $r_C \leftarrow 0$ ,  $(m_t, n_t) \leftarrow (\hat{m}, \hat{n})$  and  $accept \leftarrow \text{False}$ ;
4   Obtain the response value  $R_t(\hat{m}, \hat{n})$  using (11);
5   if  $R_t(\hat{m}_0, \hat{n}_0) < \tau_s$  then // Re-detection
6     Perform re-detection to obtain the target candidate center position set  $U_t$  using (7);
7     Obtain the target position  $(m_t, n_t)$  using Algorithm 1;
8   else if  $R_t(\hat{m}_0, \hat{n}_0) > \tau_s$  and  $t \bmod 10 = 0$  then // Re-detection
9     Perform re-detection to obtain the target candidate center position set  $U_t$  using (7);
10    Obtain the target position  $(m_t, n_t)$  and the evaluation value  $r_C$  by the C-expert using Algorithm 1;
11    if  $r_C = 1$  then
12       $accept \leftarrow \text{True}$ ;
13  if  $accept = \text{True}$  then
14    Update  $\tilde{C}_t$  with the adjusted learning rate  $\eta'$  using (5) and (6), where  $\eta = \eta'$ ;
15  else
16    Update  $\tilde{C}_t$  with the original learning rate  $\eta$  using (5) and (6);

```

3.4.2. Features

The features play an important role in visual tracking because targets may undergo rapid appearance variations during the long-term tracking process, which requires the features to be invariant to these variations. To effectively handle these challenges, both of the proposed D-expert and C-expert employ CNN features. Specifically, the features that the D-expert uses are extracted from the last fully-connected layer of the VGGNet. These features encode the semantic information of objects, and they are robust to appearance variations. Additionally, the features used in the C-expert are extracted from different convolutional layers (i.e., the conv3-4, conv4-4 and conv5-4 layers) of the VGGNet. The convolutional features are more suitable for correlation filters because they keep more spatial information, which help the correlation filters locate the target more accurately.

4. Experimental Results

We evaluate the proposed method on three challenging datasets: OTB-2013 [13], OTB-2015 [5] and UAV20L [18] datasets. The OTB-2015 dataset is one of the most widely used datasets for visual object tracking, and it contains 100 challenging video sequences. The video sequences contain various challenging factors, such as deformation, occlusion, fast motion and motion blur. More details have been discussed in [5]. The OTB-2013 dataset contains 51 video

sequences. And the OTB-2015 dataset is an extension of the OTB-2013 dataset. For completeness, we evaluate the proposed method on both the OTB-2013 and OTB-2015 datasets. To analyze the long-term tracking ability of RDLT, we evaluate RDLT on the UAV20L dataset [18] that includes 20 long video sequences to demonstrate the long-term tracking ability of RDLT. We evaluate RDLT on this dataset with the same metrics used for OTB-2013 and OTB-2015.

We compare the proposed RDLT method with 11 state-of-the-art trackers including CF2 [10], SiamFC [17], HDT [11], KCF [7], DSST [50], LCT [14], SRDCF [51], Staple [52], CSR-DCF [53], BACF [54], and MCPF [55]. Note that both CF2 and HDT are the deep correlation filter based trackers. LCT and MCPF are the two state-of-the-art long-term trackers, which achieve competitive results in the task of long-term object tracking. SiamFC is a deep Siamese network based tracker. These four methods are closely related to our method. Staple, KCF, DSST, SRDCF, CSR-DCF and BACF are hand-crafted feature based correlation filter trackers, which are proposed in recent years and have shown effectiveness for visual object tracking. For the fairness, we run the source codes of these competing trackers provided by the authors.

4.1. Experimental Setup and Evaluation Metric

Experimental Setup: For the correlation filters ($\tilde{C}_t$ and $\tilde{C}_1$) in the proposed method, the regularization parameter λ in (1) is set to 1×10^{-4} , the learning rate η in (5) is set to 1×10^{-2} and the learning ratio α is set to 10. The weights in (4) are respectively set to 5×10^{-1} , 1 and 2×10^{-2} for the conv3-4, conv4-4 and conv5-4 layers, which are similar to [10]. For the pre-trained Siamese network, we extend it with 36 searching scales. More details about the Siamese network are discussed in [17]. For the D-expert and C-expert, we set $|V_0| = |V_1| = |V_2| = 1$, $K = 1$. τ_s is experimentally set to 1×10^{-1} . The VGGNet-19 model [21] trained on the ImageNet dataset [56] is used for feature extraction. The proposed RDLT method is implemented in Matlab R2012b with MEX files in C++ and it runs at 8.9 frames per second on a computer equipped with an Intel 6700 K 4.0 GHz CPU and an NVIDIA GTX 1080 GPU.

Evaluation Metric: We follow the evaluation protocol in [5], where the distance precision (DP) plot and the overlap success (OS) plot are used as the evaluation metrics. The DP plot shows the percentage of frames in which the Euclidean distance between the estimated location and the ground-truth location is within a given threshold distance. The OS plot depicts the percentage of frames, where the overlap ratio between the predicted bounding box and the ground-truth bounding box is higher than a threshold. Additionally, the DP rate at the specific threshold of 20 pixels (DPR) is chosen as the evaluation score and the area under curve (AUC) of each OS plot is calculated for evaluating different trackers. In addition, the OS rate at the threshold of 0.5 (OSR) is also reported for more detail evaluation. We adopt the one-pass evaluation (OPE) to evaluate all the competing trackers in both the OTB and UAV20L experiments.

4.2. Evaluation on the OTB-2013 and OTB-2015 Datasets

4.2.1. Overall Performance

Figure 3 shows the overall performance obtained by the 12 competing trackers on OTB-2013 using the OPE metric. The proposed RDLT method performs favorably against the other 11 trackers on both the DP and OS metrics. As can be seen in Figure 3, the DP rate at the threshold of 20 pixels obtained by RDLT is ranked at the first. Compared with the three closely related trackers (i.e., SiamFC, CF2 and HDT), RDLT outperforms them by large margins on both the AUC scores (i.e., 8.3–8.8%) and the DP rates at 20 pixels (i.e., 2.9–14.5%). Moreover, RDLT achieves the much better overall performance than the long-term tracking method (i.e., LCT) on the OTB-2013 dataset, which can be explained that RDLT can indeed avoid some tracking failures caused by erroneous updates by taking advantage of the robust re-detection module.

We also evaluate the 12 competing trackers over all the 100 video sequences on the OTB-2015 dataset using the OPE metric, which is reported in Figure 4. As we can see, RDLT achieves competitive results in both the DP and OS plots. In the DP plot, the proposed RDLT method achieves the best DPR (88.7%) followed by MCPF (87.3%), HDT (84.8%), CF2 (83.7%) and BACF (81.7%). This comparison demonstrates the high accuracy results obtained by our method. The main reason is that RDLT not only encodes the target appearance variations, but also effectively alleviates the problem of erroneous updates by taking advantage of the pre-trained CNN model. Moreover, RDLT obtains the best OS performance when the overlap threshold varies in the range [0, 0.630]. The AUC scores obtained by the top four trackers are 63.1% (RDLT), 62.8% (MCPF), 61.6% (BACF) and 59.8% (SRDCF), respectively. It is worthy noting that RDLT uses the same scale estimation mechanism as in Staple, DSST and LCT, but RDLT still achieves the best AUC performance. The main reason is that RDLT locates the target more accurately than the other compared trackers. Compared with the deep Siamese network based tracker SiamFC, RDLT achieves the much better performance on both the DP and OS metrics by more effectively modeling the target appearance

variations during the tracking process. Overall, the proposed RDLT method achieves the best overall performance on the OTB-2015 dataset compared to the 11 state-of-the-art tracking methods.

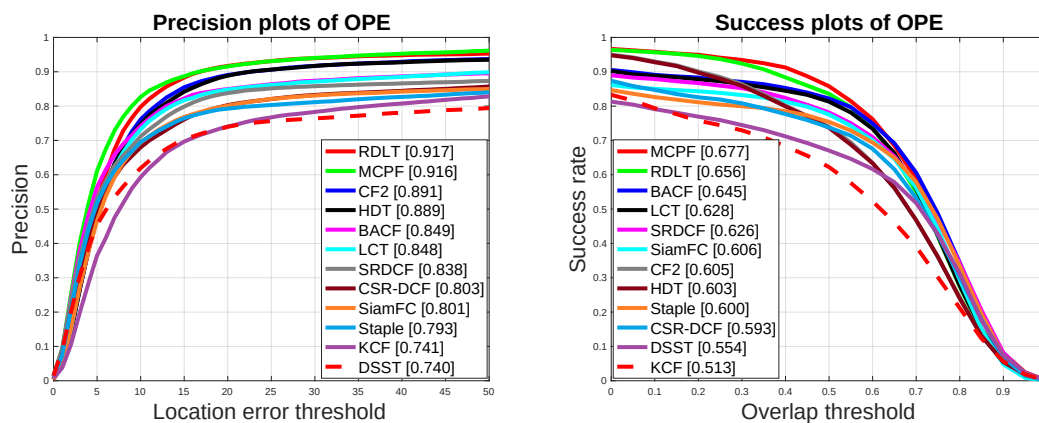


Figure 3. The overall performance of the distance precision and overlap success plots obtained by the 12 competing trackers over all the 51 video sequences on the OTB-2013 dataset using the one-pass evaluation (OPE) metric. The DPR and AUC obtained by each method are listed in the legends.

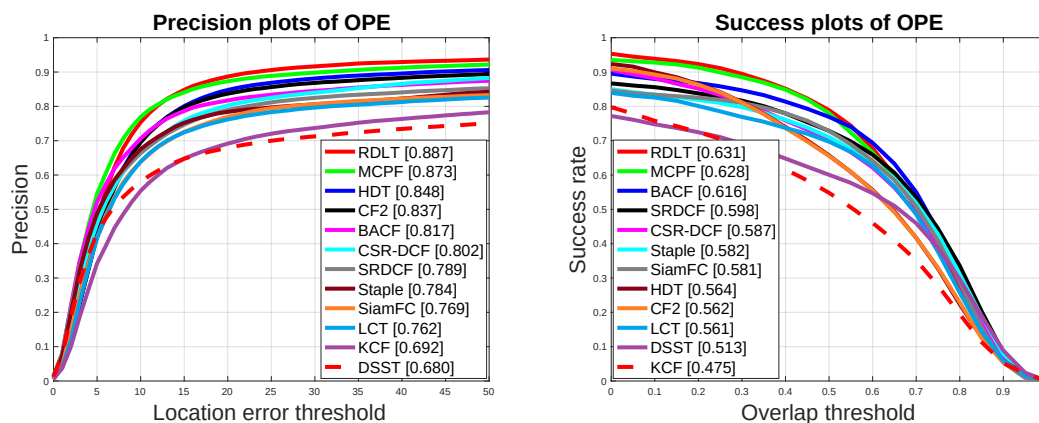


Figure 4. The overall performance of the distance precision and overlap success plots obtained by the 12 competing trackers over all the 100 video sequences on the OTB-2015 dataset using the one-pass evaluation (OPE) metric. The DPR and AUC obtained by each method are listed in the legends.

Table 1 shows the overall OPE results of the 12 competing trackers on the OTB-2013 and OTB-2015 datasets. As can be seen, the OSR obtained by RDLT on the OTB-2015 dataset is 78.9%, which is ranked at the first and outperforms the second best tracker MCPF with a relative gain of 1.2%. On the OTB-2013 dataset, RDLT achieves the comparable OSR (83.6%) to MCPF (85.8%). Meanwhile, the proposed RDLT method outperforms MCPF in terms of the DPR on both the two OTB datasets, which demonstrates the overall effectiveness of RDLT. In addition, we test the speed of each tracker and report the average tracking speed in Table 1. The average speed of RDLT is 8.9 FPS, which is about 17.8 times faster than MCPF, and it is close to the average speed of deep learning-based correlation filters (e.g., CF2 and HDT).

Table 1. Comparisons with the state-of-the-art trackers on the OTB-2013 [13] and OTB-2015 [5] datasets using the OPE metric. The average tracking speed (* indicates the GPU speed, otherwise it indicates the CPU speed) is reported in the last row. The best value is highlighted by bold.

		HDT [11]	CF2 [10]	SRDCF [51]	SiamFC [17]	Staple [52]	CSR-DCF [53]	KCF [7]	DSST [50]	BACF [54]	MCPF [55]	LCT [14]	RDLT Ours
OTB-2013	DPR (%)	88.9	89.1	83.8	80.1	79.3	80.3	74.1	74.0	84.9	91.6	84.8	91.7
	OSR (%)	73.7	74.0	78.1	77.5	75.4	73.8	62.2	67.0	82.2	85.8	81.3	83.6
OTB-2015	DPR (%)	84.8	83.7	78.9	76.9	78.4	80.2	69.2	68.0	81.7	87.3	76.2	88.7
	OSR (%)	65.7	65.5	72.8	72.8	71.0	70.2	54.8	60.1	76.9	78.0	69.7	78.9
Speed (FPS)		11.1 *	10.5 *	3.8	113.1 *	62.4	53.9	205.3	23.6	40.1	0.5	21.0	8.9 *

4.2.2. Attribute-Based Performance Analysis

The video sequences in the OTB-2015 dataset [5] are annotated with 11 attributes. The 11 attributes are background clutter, deformation, fast motion, in-plane rotation, out-of-plane rotation, low resolution, occlusion, out-of-view, scale variation, motion blur and illumination variation, respectively. To facilitate the performance analysis of all the 12 competing trackers, we evaluate each tracker using the annotated video sequences. The DP plots obtained by the 12 trackers for the video sequences with these 11 attributes using the OPE metric are shown in Figure 5. The proposed RDLT method achieves leading accuracy on the videos annotated with 10 out of 11 attributes. As a long-term tracking method, RDLT is good at handling with the problem of out-of-view that usually exists in the long-term tracking task, which demonstrates the long-term tracking ability of RDLT. Specifically, the DPR obtained by RDLT on the videos annotated with the attribute of out-of-view is 83.7%, which significantly outperforms the second best tracker MCPF with a large margin of 7.3%. This is because that RDLT maintains a robust re-detection module and a discriminative multi-expert evaluation schema, which can effectively generate reliable target candidate windows even after long-term occlusion and accurately select the optimal result from candidates, respectively.

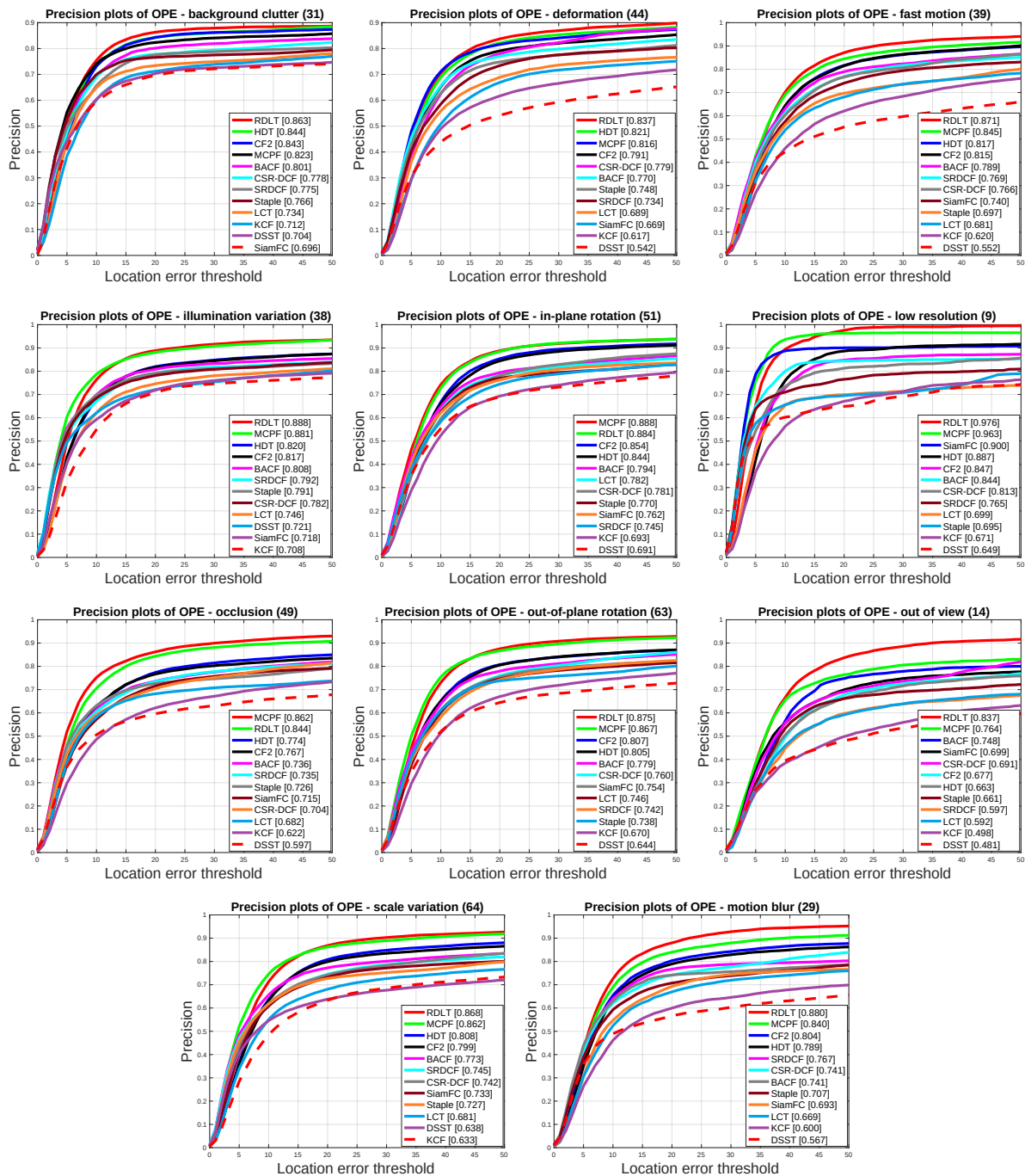


Figure 5. Attributed-based performance analysis. The DP plots obtained by the 12 trackers on the video sequences of the OTB-2015 dataset with different attributes using the OPE metric. The value in the bracket in the title of each sub-figure is the number of videos in the subset with a different attribute.

4.2.3. Qualitative Comparison

In this subsection, we qualitatively compare the proposed RDLT method with 7 state-of-the-art trackers on 10 typical video sequences with different challenges. The 7 trackers include: (i) 3 top performing deep learning based trackers on the OTB-2015 dataset (i.e., see the DP plots in Figure 4): MCPF, HDT and CF2; (ii) 3 top performing correlation filter based trackers: BACF, SRDCF and LCT; (iii) one representative deep CNN based tracker: SiamFC. The qualitative tracking results are shown in Figure 6.



Figure 6. Some qualitative tracking results obtained by the proposed RDLT method and 7 state-of-the-art trackers on 10 video sequences with different challenges. The 10 video sequences are respectively the (a) Box, (b) Girl2, (c) Bird1, (d) Biker, (e) BlurCar1, (f) Soccer, (g) MotorRolling, (h) CarDark, (i) Skiing, and (j) Jump video sequences.

Long-term occlusion and out-of-view: Figure 6a–c show the qualitative tracking results obtained by the proposed RDLT method and other competing methods on the Box, Girl2 and Bird1 sequences, where the targets in these sequences suffer from long-term occlusion and out-of-view. In the sequences of Box and Girl2, the box and the girl are occluded at frame #463 and #123, respectively. When the targets reappear, all the trackers except for the proposed RDLT and MCPF drift and cannot successfully track the box at the subsequent frames. Compared with MCPF, the proposed RDLT achieves better OS performance. In addition, we can see that the long-term correlation tracker (LCT) cannot recover from the tracking failures after long-term occlusion, which can be attributed to the erroneous updates of its re-detection module. In contrast, the re-detection module in RDLT is trained in an offline manner, and it has no risks of erroneous updates of the detection module. Moreover, the multi-expert evaluation schema in RDLT effectively evaluates the target candidate windows generated by the detector, by which RDLT takes the optimal target window as the final result. In the Bird1 sequence, the bird gets close to the cloud and then it is occluded by the cloud very soon. It undergoes out-of-view from frame #124 to frame #150. For most of the correlation filter-based trackers (e.g., CF2, SRDCF, BACF, LCT and HDT), they cannot track the bird when it reappears at frame #250 for two reasons: A) they do not have an effective re-detection module to re-detect the bird when it reappears and b) they erroneously update the appearance model of the bird during the long-term occlusion. In comparison, the proposed RDLT method can successfully track the bird and effectively handle the challenges of long-term occlusion and out-of-view.

Fast motion and motion blur: We evaluate various trackers on the Biker, BlurCar1 and Soccer sequences with the challenges of fast motion and motion blur. The qualitative results are shown in Figure 6d–f. For the Biker sequence, the man turns round quickly from frame #70 to frame #81, where the target simultaneously undergoes fast motion and motion blur. We can find that all the trackers lose the man except for RDLT and SiamFC. In comparison, our RDLT performs more accurate target scale estimation due to its multi-scale searching strategy. For the BlurCar1

sequence, the car undergoes motion blur at frames #763 and #890. RDLT accurately tracks the car in the whole process, which shows its ability in dealing with motion blur. In the Soccer sequence, the football player suffers from motion blur at frames #166 and #198. The deep CNN based trackers including CF2, HDT, MCPF and our RDLT, achieve favorably tracking results in these frames, which can be attributed to the powerfulness of the CNN features used in these trackers.

Rotation and illumination variation: Figure 6g shows the tracking results on the MotorRolling sequence with rotation. In the sequence, the motorbike rotates from frame #10 to frame #164. The hand-crafted feature based trackers (i.e., BACF and SRDCF) gradually lose the target. In comparison, several deep feature based correlation filter trackers (i.e., HDT, CF2, MCPF and RDLT) successfully track the motorbike in the whole process, which is mainly due to the rotation invariance of the CNN features they use. Note that RDLT achieves better OS performance than the other three deep feature based correlation filter trackers (i.e., CF2, MCPF and HDT) due to its better localization ability. In Figure 6h, for the CarDark sequence, the car suffers from illumination variations. By using the CNN features that are invariant to illumination variations, the proposed RDLT achieves the best OS performance and tracks the car well in the whole process.

Deformation and scale variation: Figure 6i,j illustrates the tracking results on the Skiing and Jump sequences, respectively, where the targets undergo deformation and significant scale variations. In the Skiing sequence, the man undergoes large scale variations. RDLT tracks the man accurately in the whole process and estimates the scale of the man more accurately than the other trackers. For the Jump sequence, it is challenging because the athlete jumps with significant deformation and large scale variations. Although RDLT loses the athlete at several frames, it successfully re-detects the athlete at frame #100, while all the other trackers lose the athlete, which demonstrates the effectiveness of the re-detection module in RDLT.

4.3. Evaluation on the UAV20L dataset

The UAV20L dataset consists of 20 long video sequences, which include various challenges (e.g., occlusion and fast motion) that are far from being fully solved in the current vision systems of intelligent vehicles. To further evaluate the long-term tracking ability of RDLT and measure its application in intelligent vehicles, we evaluate RDLT on the UAV20L [18] dataset with the comparison to 8 state-of-the-art trackers including MCPF [55], KCFDP [57], HCF [10], HDT [11], MUSTER [9], Staple [52], MEEM [58] and SRDCF [51].

Figure 7 illustrates the DP and OS plots obtained by all the 9 competing trackers on the UAV20L dataset. As can be seen, the proposed RDLT achieves the comparable DPR (57.4%) to MCPF (58.8%), which significantly outperforming the other trackers with large margins. In terms of the AUC metric, our RDLT achieves the best result (39.5%) and outperforms the second ranked tracker (MCPF) with a relative gain of 8.5%, which shows the good long-term tracking ability of RDLT. Compared with the baseline tracker (HCF), our RDLT outperforms it with large margins of 8.4% and 4.3% in terms of the DPR and AUC metrics, respectively. This large improvements can be attributed to the robust re-detection module and multi-expert evaluation schema used in the proposed RDLT. Overall, the proposed RDLT achieves the favorably performance on the UAV20L dataset and shows that it has the great potential for applications to intelligent vehicles.

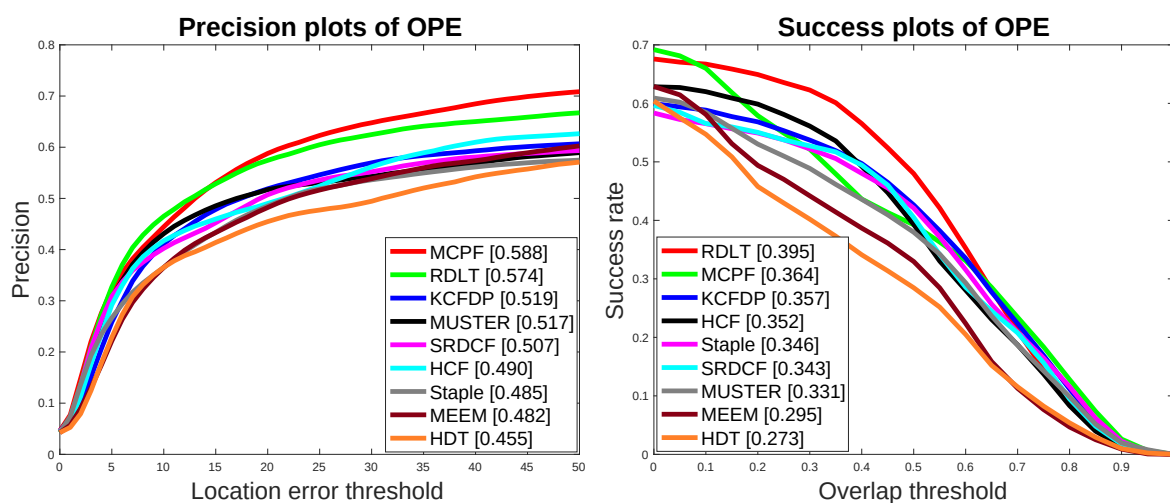


Figure 7. The overall performance of the precision and success plots obtained by the 9 evaluated trackers over all the 20 long video sequences on the UAV20L dataset using the one-pass evaluation (OPE) metric. The DPR and AUC obtained by each method are listed in the legends.

4.4. Comparison with Deep Siamese/Transformer-Based Trackers

To further compare our method with modern deep similarity-based trackers, we provide a dedicated comparison on the OTB-100 dataset between representative deep Siamese/Transformer-based trackers and classical correlation filter (CF) based trackers in Table 2. The two categories differ fundamentally in their training paradigm: the deep Siamese/Transformer-based trackers (e.g., SiamFC [17], DiMP [16], TransT [28], STARK [29] and DropTrack [36]) are trained offline on large-scale tracking datasets (e.g., LaSOT, TrackingNet, GOT-10k and COCO) and do not perform any online model updating during tracking; in contrast, the CF-based trackers do not rely on such large-scale offline tracking data and instead adapt the target appearance model through online updating. The proposed RDLT method, although equipped with a pre-trained Siamese re-detector, fundamentally belongs to the correlation filter family because its base tracker and target appearance model are still online-updated correlation filters, and it is therefore listed in the CF category.

Table 2. Comparison between deep Siamese/Transformer-based trackers and correlation filter (CF) based trackers on the OTB-100 dataset in terms of AUC (%). All reported values are taken from the original publications and from our experiments.

Category	Method	OTB-100 AUC (%)
Deep Siamese/Transformer-based	SiamFC [17]	58.1
	DiMP [16]	53.7
	TransT [28]	69.5
	STARK [29]	68.1
	DropTrack [36]	69.6
Correlation filter based	KCF [7]	47.5
	DSST [50]	51.3
	LCT [14]	56.1
	CF2 [10]	56.2
	HDT [11]	56.4
	Staple [52]	58.2
	CSR-DCF [53]	58.7
	SRDCF [51]	59.8
	BACF [54]	61.6
	MCPF [55]	62.8
	RDLT (Ours)	63.1

As shown in Table 2, recent Transformer-based trackers such as TransT, STARK and DropTrack achieve higher AUC scores on OTB-100 than the classical CF-based trackers, which is a phenomenon widely observed in the tracking community. This performance gap mainly comes from the fact that these modern trackers are fully data-driven and benefit from very large-scale offline training, which allows them to learn highly discriminative target representations. In contrast, the main advantage of correlation filter-based trackers is that they can naturally perform online updating of the target appearance model during tracking, and therefore can adapt to target appearance variations without heavily relying on large-scale offline training. Our RDLT method is intentionally designed to take advantage of both worlds: the target candidate window generation step in RDLT is decoupled from the specific Siamese model (SiamFC) used inside it, so the framework is flexible and the SiamFC re-detector can be replaced by any more modern Siamese/Transformer tracker (e.g., TransT or DropTrack) to further improve the quality of the candidate windows, while the online-updated correlation filter and the multi-expert evaluation schema continue to provide robustness against erroneous updates.

4.5. Speed Analysis

As reported in Table 1, the proposed RDLT runs at 8.9 FPS on a single NVIDIA GTX 1080 GPU, which is lower than several short-term trackers such as SiamFC, KCF and Staple. After a careful re-examination of each module of RDLT, we identify that the main runtime bottleneck of the current RDLT implementation is the base correlation filter tracker itself, namely CF2 [10], which only runs at about 10.5 FPS in our setting. Since RDLT calls the base tracker at every frame, the overall speed of RDLT is intrinsically bounded by the speed of the underlying CF tracker. Importantly, the proposed RDLT framework is flexible with respect to the choice of the base CF tracker: the multi-expert evaluation schema and the Siamese re-detection module do not depend on the specific implementation of the base CF, so the current CF2 base tracker can be readily replaced by more efficient

correlation filter trackers (e.g., KCF [7], Staple [52], BACF [54] or CSR-DCF [53], whose FPS in Table 1 ranges from 40 to 200+). With a faster base CF tracker, the overall speed of RDLT can be significantly improved to satisfy the high-speed requirements of real-world applications such as on-board tracking for intelligent vehicles and drones.

4.6. Ablation Study

To verify the individual contribution of each core component of RDLT, namely (i) the deep Siamese re-detection module and (ii) the multi-expert evaluation schema, we evaluate four variants on both the OTB-2013 and OTB-2015 datasets and report the OSR (%) (i.e., the overlap success rate at the threshold of 0.5) in Table 3. The four variants are: (V1) the base correlation filter tracker alone, without any Siamese re-detection or multi-expert evaluation; (V2) the pre-trained SiamFC re-detector used independently as a tracker; (V3) the base correlation filter equipped with the deep Siamese re-detector, where the optimal target candidate window is selected by the D-expert only; and (V4) the full RDLT, where the optimal candidate is jointly selected by the D-expert and the C-expert.

Table 3. Ablation study of the proposed RDLT method on the OTB-2013 and OTB-2015 datasets in terms of OSR (%). The four variants progressively add the Siamese re-detection module and the multi-expert evaluation components on top of the base correlation filter.

Variant	Configuration	OTB-2013	OTB-2015
V1	Base correlation filter [10] (no re-detection, no expert)	74.0	65.5
V2	SiamFC [17] re-detector alone	77.5	72.8
V3	Base CF [10] + Siamese [17] re-detection + D-expert only	78.1	73.2
V4	Full RDLT: Base CF + Siamese re-detection + D-expert + C-expert	83.6	78.9

From Table 3, we first observe that using the correlation filter alone (V1) or the pre-trained Siamese network alone (V2) yields clearly inferior tracking performance. The base correlation filter (V1) suffers from the well-known error-accumulation problem during online updating and therefore quickly drifts once the target undergoes long-term occlusion or out-of-view. The pre-trained SiamFC re-detector alone (V2), although free of online-update errors, relies only on the fixed offline-learned template and is therefore very sensitive to significant target appearance variations along the video. Neither component is sufficient by itself for robust long-term tracking. Building on this observation, the key contribution of the proposed multi-expert evaluation schema is to make the online-updated correlation filter and the offline-trained Siamese re-detector work cooperatively through two complementary experts. As shown by V1 → V3, equipping the base correlation filter with the Siamese re-detector and the D-expert already brings a clear improvement (74.0 → 78.1 on OTB-2013 and 65.5 → 73.2 on OTB-2015), which confirms that using the deep similarity score from the offline-learned target template to select the optimal candidate effectively rescues the tracker from drifting. However, the D-expert alone is anchored to the initial target template and may select a sub-optimal window once the target’s appearance has substantially evolved. The C-expert is designed exactly to complement this limitation: it scores each candidate using the response of the online-adapted correlation filter, which captures the most recent target appearance and is highly discriminative against the surrounding background. As shown by V3 → V4, jointly using the D-expert and the C-expert produces a substantial further gain (78.1 → 83.6 on OTB-2013 and 73.2 → 78.9 on OTB-2015), and the full RDLT (V4) significantly outperforms both single-component baselines. This clearly verifies that the two experts are complementary—the offline-learned D-expert anchors the candidate selection against erroneous updates, while the online-updated C-expert keeps the selection aware of the latest target appearance—and that jointly using both experts is essential for achieving more robust target candidate selection during online long-term tracking.

4.7. Failure-Case Analysis

While the qualitative results above demonstrate that RDLT performs robustly on a wide range of challenging scenarios, we further analyze the typical failure cases of the proposed method. We identify three representative scenarios in which RDLT may fail: (i) when multiple objects with very similar appearance appear simultaneously in the search region (visual distractors), the D-expert may incorrectly select a distractor as the target since its decision is anchored to the offline-learned template; (ii) when the target is at very low resolution over a long period, both the C-expert and the D-expert may suffer from feature degradation, leading to inaccurate localization; and (iii) when the target undergoes extreme scale change combined with heavy deformation, the multi-scale Siamese search may not provide sufficiently accurate candidate windows. We expect to address these failure modes in future work, e.g.,

through distractor-aware matching, resolution-adaptive feature extraction and more advanced Transformer-based re-detection modules (e.g., TransT [28] or DropTrack [36]).

5. Conclusions

In this paper, we propose a robust deep long-term tracking method (called RDLT) that is capable of coping well with the challenges of long-term occlusion and out-of-view, by embedding a pre-trained deep Siamese network into a unified correlation filter tracking framework. In the proposed method, we present a novel multi-expert evaluation schema, which includes a correlation filter-based expert (C-expert) and a deep similarity-based expert (D-expert), to effectively evaluate all the target candidate windows generated by both a base tracker and a target detector. Both experts are combined to select the optimal target candidate window as the tracked target. Due to the effectiveness of the multi-expert evaluation schema, the correlation filter in the proposed RDLT can avoid being greatly affected by corrupt samples. Thus, the proposed RDLT method is robust to handle with erroneous updates, which are mainly caused by long-term occlusion and out-of-view. Extensive experiments on three popular datasets demonstrate that the proposed RDLT performs favorably against several state-of-the-art trackers, and it is particularly good at handling long-term occlusion and out-of-view for long-term tracking, which shows its great potential for autonomous navigation to intelligent robots and vehicles.

6. Future Work

Although the proposed RDLT method has shown promising long-term tracking performance, several challenges remain open when deploying it in real-world intelligent vehicles and drones, and we plan to address them along the following directions. First, on-board real-time deployment with limited computation: as discussed in the Speed Analysis subsection, the runtime of RDLT is mainly bounded by the base correlation filter tracker, and we will explore lightweight base CF trackers (e.g., KCF [7], Staple [52] or BACF [54]) together with GPU re-implementation of the multi-expert evaluation schema to satisfy the strict latency requirements of in-vehicle and on-drone perception. Second, robustness to severe weather, motion blur and low-light conditions typical of driving scenes: the current pre-trained Siamese re-detector is mostly trained on standard tracking benchmarks, and we will study domain adaptation and data augmentation strategies (e.g., rain/fog synthesis and low-light enhancement) to make RDLT more robust in adverse environments. Third, domain adaptation across ground-view (driving) and aerial-view (drone) footage: although our evaluation on UAV20L shows encouraging long-term tracking ability in the aerial view, the gap between ground-view and aerial-view targets in scale, viewpoint and motion pattern is still non-trivial; we will explore cross-view feature alignment and unsupervised domain adaptation to improve the cross-platform generalization of RDLT. Finally, we will also explore replacing the SiamFC re-detector inside RDLT with more modern Transformer-based trackers (e.g., TransT [28] or DropTrack [36]) to further improve the quality of the target candidate windows while keeping the online-updated correlation filter and the multi-expert evaluation schema for long-term robustness.

Author Contributions

H.D.: writing, investigation, and editing; X.G.: reviewing and editing; Q.W.: methodology and conceptualization. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Data are contained within the article.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used ChatGPT to improve the language and readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Rakha, H.A.; Ahn, K.; Faris, W.; et al. Simple Vehicle Powertrain Model for Modeling Intelligent Vehicle Applications. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 770–780. <https://doi.org/10.1109/tits.2012.2188517>.
2. Balid, W.; Tafish, H.; Refai, H.H. Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1784–1794. <https://doi.org/10.1109/tits.2017.2741507>.
3. Cao, Z.; Li, J.; Shao, S.; et al. Siamese Adaptive Network-Based Accurate and Robust Visual Object Tracking Algorithm for Quadrapedal Robots. *IEEE Trans. Cybern.* **2025**, *55*, 1264–1276. <https://doi.org/10.1109/tcyb.2025.3527342>.
4. Cao, Z.; Li, J.; Zhang, D.; et al. A Multi-Object Tracking Algorithm With Center-Based Feature Extraction and Occlusion Handling. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4464–4473. <https://doi.org/10.1109/tits.2022.3229978>.
5. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. <https://doi.org/10.1109/tpami.2014.2388226>.
6. Ross, D.A.; Lim, J.; Lin, R.S.; et al. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. <https://doi.org/10.1007/s11263-007-0075-7>.
7. Henriques, J.F.; Caseiro, R.; Martins, P.; et al. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. <https://doi.org/10.1109/tpami.2014.2345390>.
8. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. <https://doi.org/10.1109/tpami.2011.239>.
9. Hong, Z.; Chen, Z.; Wang, C.; et al. MUlti-Store Tracker (MUSTer): A Cognitive Psychology Inspired Approach to Object Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 749–758. <https://doi.org/10.1109/cvpr.2015.7298675>.
10. Ma, C.; Huang, J.B.; Yang, X.; et al. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3074–3082. <https://doi.org/10.1109/iccv.2015.352>.
11. Qi, Y.; Zhang, S.; Qin, L.; et al. Hedged Deep Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4303–4311. <https://doi.org/10.1109/cvpr.2016.466>.
12. Danelljan, M.; Häger, G.; Khan, F.S.; et al. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 7–13 December 2015; pp. 621–629. <https://doi.org/10.1109/iccvw.2015.84>.
13. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418. <https://doi.org/10.1109/cvpr.2013.312>.
14. Ma, C.; Yang, X.; Zhang, C.; et al. Long-Term Correlation Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5388–5396. <https://doi.org/10.1109/cvpr.2015.7299177>.
15. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429. <https://doi.org/10.1109/cvpr.2016.158>.
16. Zhang, L.; Han, H.; Zhou, M.; et al. An Improved Discriminative Model Prediction Approach to Real-Time Tracking of Objects With Camera as Sensors. *IEEE Sens. J.* **2021**, *21*, 17308–17317. <https://doi.org/10.1109/jsen.2021.3079214>.
17. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; et al. Fully-Convolutional Siamese Networks for Object Tracking. In *Computer Vision, Proceedings of the ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 850–865. https://doi.org/10.1007/978-3-319-48881-3_56.
18. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In *Computer Vision, Proceedings of the ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 445–461. https://doi.org/10.1007/978-3-319-46448-0_27.
19. Bolme, D.; Beveridge, J.R.; Draper, B.A.; et al. Visual Object Tracking Using Adaptive Correlation Filters. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. <https://doi.org/10.1109/cvpr.2010.5539960>.
20. Henriques, J.F.; Caseiro, R.; Martins, P.; et al. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *Computer Vision, Proceedings of the ECCV 2012, Florence, Italy, 7–12 October 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715. https://doi.org/10.1007/978-3-642-33765-9_50.
21. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556v6.

22. Zhang, K.; Zhang, L.; Yang, M.H. Real-Time Compressive Tracking. In *Computer Vision, Proceedings of the ECCV 2012, Florence, Italy, 7–12 October 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 864–877. https://doi.org/10.1007/978-3-642-33712-3_62.
23. Chaudhuri, K.; Freund, Y.; Hsu, D. A Parameter-Free Hedging Algorithm. *arXiv* **2009**, arXiv:0903.2851v2.
24. Cui, Y.; Guo, D.; Shao, Y.; et al. Joint Classification and Regression for Visual Tracking with Fully Convolutional Siamese Networks. *Int. J. Comput. Vis.* **2022**, *130*, 550–566. <https://doi.org/10.1007/s11263-021-01559-4>.
25. Chen, Z.; Zhong, B.; Li, G.; et al. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6667–6676. <https://doi.org/10.1109/cvpr42600.2020.00670>.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; et al. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
28. Chen, X.; Yan, B.; Zhu, J.; et al. Transformer Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8122–8131. <https://doi.org/10.1109/cvpr46437.2021.00803>.
29. Yan, B.; Peng, H.; Fu, J.; et al. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10428–10437. <https://doi.org/10.1109/iccv48922.2021.01028>.
30. Wang, N.; Zhou, W.; Wang, J.; et al. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1571–1580. <https://doi.org/10.1109/cvpr46437.2021.00162>.
31. Cui, Y.; Jiang, C.; Wang, L.; et al. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13598–13608. <https://doi.org/10.1109/cvpr52688.2022.01324>.
32. Ye, B.; Chang, H.; Ma, B.; et al. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 341–357.
33. Mayer, C.; Danelljan, M.; Bhat, G.; et al. Transforming Model Prediction for Tracking. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8721–8730. <https://doi.org/10.1109/cvpr52688.2022.00853>.
34. Lin, L.; Fan, H.; Zhang, Z.; et al. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 16743–16754.
35. Chen, X.; Peng, H.; Wang, D.; et al. SeqTrack: Sequence to Sequence Learning for Visual Object Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 14572–14581. <https://doi.org/10.1109/cvpr52729.2023.01400>.
36. Wu, Q.; Yang, T.; Liu, Z.; et al. DropMAE: Masked Autoencoders with Spatial-Attention Dropout for Tracking Tasks. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 14561–14571. <https://doi.org/10.1109/cvpr52729.2023.01399>.
37. Wu, Q.; Yang, T.; Fang, B.; et al. M2P: Improving Visual Foundation Models with Mask-to-Point Weakly-Supervised Learning for Dense Point Tracking. *arXiv* **2026**, arXiv:2603.17813v1.
38. Wu, Q.; Chan, A.B. Learning Tracking Representations from Single Point Annotations. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 17–18 June 2024; pp. 2606–2615. <https://doi.org/10.1109/cvprw63382.2024.00267>.
39. Wu, Q.; Wan, J.; Chan, A.B. Progressive Unsupervised Learning for Visual Object Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2992–3001. <https://doi.org/10.1109/cvpr46437.2021.00301>.
40. Wu, Q.; Yang, T.; Wu, W.; et al. Scalable Video Object Segmentation with Simplified Framework. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 13833–13843. <https://doi.org/10.1109/iccv51070.2023.01276>.
41. Wu, Q.; Yan, Y.; Liang, Y.; et al. DSNet: Deep and Shallow Feature Learning for Efficient Visual Tracking. In *Computer Vision, Proceedings of the ACCV 2018, Perth, WA, Australia, 2–6 December 2018*; Springer International Publishing: Cham, Switzerland, 2019; pp. 119–134. https://doi.org/10.1007/978-3-030-20873-8_8.
42. Wu, Q.; Chan, A.B. Meta-Graph Adaptation for Visual Object Tracking. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6. <https://doi.org/10.1109/icme51207.2021.9428441>.
43. Wei, X.; Bai, Y.; Zheng, Y.; et al. Autoregressive Visual Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 9697–9706. <https://doi.org/10.1109/cvpr52729.2023.00935>.

44. Bai, Y.; Zhao, Z.; Gong, Y.; et al. ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 19048–19057. <https://doi.org/10.1109/cvpr52733.2024.01802>.
45. Zheng, Y.; Zhong, B.; Liang, Q.; et al. ODTrack: Online Dense Temporal Token Learning for Visual Tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 7588–7596.
46. Xie, J.; Zhong, B.; Mo, Z.; et al. Autoregressive Queries for Adaptive Tracking with Spatio-Temporal Transformers. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 19300–19309. <https://doi.org/10.1109/cvpr52733.2024.01826>.
47. Kristan, M.; Matas, J.; Danelljan, M.; et al. The First Visual Object Tracking Segmentation VOTS2023 Challenge Results. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2–6 October 2023; pp. 1788–1810. <https://doi.org/10.1109/iccvw60793.2023.00195>.
48. Kristan, M.; Matas, J.; Tokmakov, P.; et al. The Second Visual Object Tracking Segmentation VOTS2024 Challenge Results. In *Computer Vision, Proceedings of the ECCV 2024 Workshops, Milan, Italy, 29 September–4 October 2024*; Springer Nature: Cham, Switzerland, 2025; pp. 357–383. https://doi.org/10.1007/978-3-031-91767-7_24.
49. Wu, Q.; Yu, Y.; Kong, C.; et al. Temporal Unlearnable Examples: Preventing Personal Video Data from Unauthorized Exploitation by Object Tracking. In Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV), Honolulu, HI, USA, 19–25 October 2025; pp. 11110–11121. <https://doi.org/10.1109/iccv51701.2025.01034>.
50. Danelljan, M.; Häger, G.; Khan, F.S.; et al. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, 39, 1561–1575. <https://doi.org/10.1109/tpami.2016.2609928>.
51. Danelljan, M.; Häger, G.; Khan, F.S.; et al. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318. <https://doi.org/10.1109/iccv.2015.490>.
52. Bertinetto, L.; Valmadre, J.; Golodetz, S.; et al. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409. <https://doi.org/10.1109/cvpr.2016.156>.
53. Lukežič, A.; Vojšić, T.; Čehovin Zajc, L.; et al. Discriminative Correlation Filter with Channel and Spatial Reliability. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856. <https://doi.org/10.1109/cvpr.2017.515>.
54. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1144–1152. <https://doi.org/10.1109/iccv.2017.129>.
55. Zhang, T.; Xu, C.; Yang, M.H. Multi-Task Correlation Particle Filter for Robust Object Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4819–4827. <https://doi.org/10.1109/cvpr.2017.512>.
56. Deng, J.; Dong, W.; Socher, R.; et al. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. <https://doi.org/10.1109/cvpr.2009.5206848>.
57. Huang, D.; Luo, L.; Chen, Z.; et al. Applying Detection Proposals to Visual Tracking for Scale and Aspect Ratio Adaptability. *Int. J. Comput. Vis.* **2017**, 122, 524–541. <https://doi.org/10.1007/s11263-016-0974-6>.
58. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 188–203. https://doi.org/10.1007/978-3-319-10599-4_13.