


Article

Robust Visual Tracking with Reliable Object Information and Kalman Filter

Hang Chen ^{*}, Weiguo Zhang and Danghui Yan

Automation College, Northwestern Polytechnical University, Xi'an 710072, China; zhangwg@nwpu.edu.cn (W.Z.); yandh@mail.nwpu.edu.cn (D.Y.)

* Correspondence: xgdhang@mail.nwpu.edu.cn

Abstract: Object information significantly affects the performance of visual tracking. However, it is difficult to obtain accurate target foreground information because of the existence of challenging scenarios, such as occlusion, background clutter, drastic change of appearance, and so forth. Traditional correlation filter methods roughly use linear interpolation to update the model, which may lead to the introduction of noise and the loss of reliable target information, resulting in the degradation of tracking performance. In this paper, we propose a novel robust visual tracking framework with reliable object information and Kalman filter (KF). Firstly, we analyze the reliability of the tracking process, calculate the confidence of the target information at the current estimated location, and determine whether it is necessary to carry out the online training and update step. Secondly, we also model the target motion between frames with a KF module, and use it to supplement the correlation filter estimation. Finally, in order to keep the most reliable target information of the first frame in the whole tracking process, we propose a new online training method, which can improve the robustness of the tracker. Extensive experiments on several benchmarks demonstrate the effectiveness and robustness of our proposed method, and our method achieves a comparable or better performance compared with several other state-of-the-art trackers.

Keywords: visual object tracking; correlation filter; reliable information; Kalman filter



Citation: Chen, H.; Zhang, W.; Yan, D. Robust Visual Tracking with Reliable Object Information and Kalman Filter. *Sensors* **2021**, *21*, 889. <https://doi.org/10.3390/s21030889>

Received: 31 December 2020

Accepted: 25 January 2021

Published: 28 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual object tracking is one of fundamental problems in the field of computer vision. This task aims to estimate the target location in all frames after the initial frame target is given. It has been widely used in many aspects of real life, including video surveillance [1], human-computer interaction [2], robots [3] and automatic drive [4]. In recent years, this field has attracted a large number of researchers and a lot of excellent works [5,6] have also emerged. Although great progress has been made in visual tracking recently, visual object tracking is still an open problem in the field of computer vision because of the challenging scenarios (e.g., deformation, illumination variation, occlusion, background clutter, etc.) in tracking process.

Recently, correlation filter (CF) [7–10] methods have attracted a lot of attention, which have the advantages of accurate tracking precision and high tracking frame rate. CF methods use cyclic shift to approximate dense sampling, which greatly increases the number of training samples, solves the problem of training samples shortage. Additionally, according to the convolution theorem, convolution operation of correlation filter is converted to frequency domain for calculation, which greatly reduces the computational complexity and enhances the real-time performance. Although the CF tracking methods have these advantages, there are still some drawbacks. Most CF methods adopt simple linear interpolation to update the model, which will lead to two problems. First, the reliability of tracking results are not analyzed. When facing challenging scenarios (e.g., occlusion, background clutter, aspect ratio change, etc.), the noise information is gradually added to the filter when online training and updating in tracking process and the model will be distorted. Second,

the first frame information in the whole tracking process is the most reliable. However, with the updating process, the first frame information in the model gradually decreases, which reduces the robustness of the model. In addition, most CF methods do not consider the relationship between frames.

In order to address these problems, we propose a robust visual tracking framework based on reliable object information and Kalman filter. The method in this paper mainly includes three modules: tracking results reliability analysis (TRRA) module, Kalman filter (KF) module and reliable online training (ROT) module. As for the noise interference problems in online training, the TRRA module will analyze the tracking results and select the most reliable object information for model training to reduce the impact of noise. For the problem of the decline of the object information in the first frame of the model, we propose a new model training method, which uses the first frame and the current frame jointly to train the model to improve the stability of the model. Finally, we use the KF module to model the object motion information, so as to supplement the CF tracking and improve the tracking robustness. Figure 1 shows that our tracker can handle complex tracking scenarios and has better tracking performance than the basic tracker CFNet [11].

The main contributions of this method can be summarized as follows:

- We propose a new reliable online training method, which can preserve the useful first frame target information.
- We develop a Kalman filter to describe the object's motion information, then use the trajectory information to guide the tracking process.
- We propose a reliability analysis method for tracking process. This ensures the validity of the target information in the model training process.
- Extensive experiments are conducted on several benchmark datasets. The results show the effectiveness and robustness of the proposed method. In addition, our method achieves a competitive tracking performance compared with other state-of-the-art trackers.

This paper is organized into 5 sections. Some related works are summarized in Section 2 and the proposed method in this paper is described in Section 3. The experimental results are provided in Section 4. Section 5 is the conclusion of this paper.



Figure 1. Tracking results in challenging scenarios. The first column is the initial frame, in which the red box object specifies the target to be tracked. The following columns are the tracking results in complex scenes, and the blue box represents the basic tracker, and the green box represents ours.

2. Related Work

2.1. Correlation Filter Methods

The method based on correlation filter was pioneered by Bolme et al. in MOSSE [12]. MOSSE is a linear discriminant classifier based on single-channel pixel feature, and achieves the frame rate over 600 FPS. Many improved CF methods have also been proposed subsequently. KCF [7,13] introduces the kernel technique into the CF methods to improve the discriminative ability of the classifier. Multi-channel features also greatly improve the tracking performance of CF methods, such as KCF uses Histogram of Oriented Gradient (HOG) features, SAMF [14] uses HOG and CN features. DSST [15] uses a one-dimensional correlation filter and multi-scale template to accurately estimate the target scale, which solves the problem of target scale variations and wins the championship on VOT2014. In order to solve the problem of training correlation filters limited in small search area, SRDCF [8] adds a space penalty term to the optimization objective function, which enables the filter to track the target in a larger searching area and reduces the boundary effect of correlation operation. With the introduction of deep convolution features by DeepSRDCF [16], the performance of SRDCF tracker has been further improved. C-COT [10] learns discriminative convolution operators and obtains confidence map of the target all in continuous space domain, to improve the richness of model and the localization accuracy. In consideration of the great influence of background information on tracking performance, Mueller et al. [17] propose a tracking framework to explicitly learn the background information around the target on CF trackers. This framework can be widely used in CF trackers to improve the tracking performance. Bibi et al. [18] proposed an adaptive target response framework, which can adaptively change the target response frame by frame, making the tracker insensitive to error locations. Xia et al. [19] build a tracker with fused deep features and correlation filters to solve challenge situations.

2.2. Deep Learning Methods

Recently, deep learning framework have been used in the field of visual tracking. Since deep learning has the characteristics of large training data sets and computational requirements, the trackers based on deep learning can be divided into two categories. One is to use convolutional neural network (CNN) pretrained on other data sets as feature extractor, and then combine with traditional methods to achieve object tracking. As mentioned in the previous subsection, DeepSRDCF [16], C-COT [10] and ECO [20] combine the deep features extracted by pretrained CNN with CF, and achieve the state-of-the-art tracking performance. The other is to fully adopt deep learning structure, and then train the tracker end-to-end on large data sets. MDNet [21] proposes a multi-domain learning model based on CNN, which can separate the independent information of multiple targets from the target. GOTURN [22] uses the image pairs of the previous frame target and the current frame search area as input, and then directly regresses the position of the target in the search area through the deep network. It can achieve the tracking frame rate of 100FPS. SINT [23] and SiamFC [24] formulate the visual tracking as a similarity learning problem. By training a similarity matching network on the detection dataset, the target in the first frame is compared with the candidate regions of the subsequent frame to realize the target estimating. There is no model updating in the tracking process, so they achieve both high frame rate and high tracking accuracy. The backbone used in SiamFC is relatively shallow, SiamDW [25] and SiamRPN++ [26] explore deeper networks to improve tracking performance. CFNet [11] takes CF as a differentiable layer of deep neural network to realize the end-to-end training of the network. SANT [27] presents structure-attention networks to learn robust structure information of targets. HKSiamFC [28] adopts Histogram model to explore target's prior color information, and makes SiamFC more robust in some complex environments.

2.3. Temporal Stability

Making full use of temporal information is very important for the robustness of visual object tracking. Many methods using temporal information are proposed to improve tracking performance. One kind of tracker simply uses temporal information, such as CF [7,8,13] and some deep trackers [11,23,24], by focusing on the region near the target in the previous frame and suppressing other remote regions. The other kind of tracker is to encode temporal information directly by Recurrent Neural Network (RNN) [29] or Long Short-Term Memory (LSTM) [30,31]. In this paper, we use KF to model the object motion, and then use the temporal information of video sequence to supplement the tracking process.

3. Proposed Method

In this section, we will elaborate on the method proposed in this paper, which mainly includes three components: reliability analysis, Kalman filter, and reliable online training method. Finally, we also introduce the filter update and tracking details. The overview of our proposed method is shown in Figure 2.

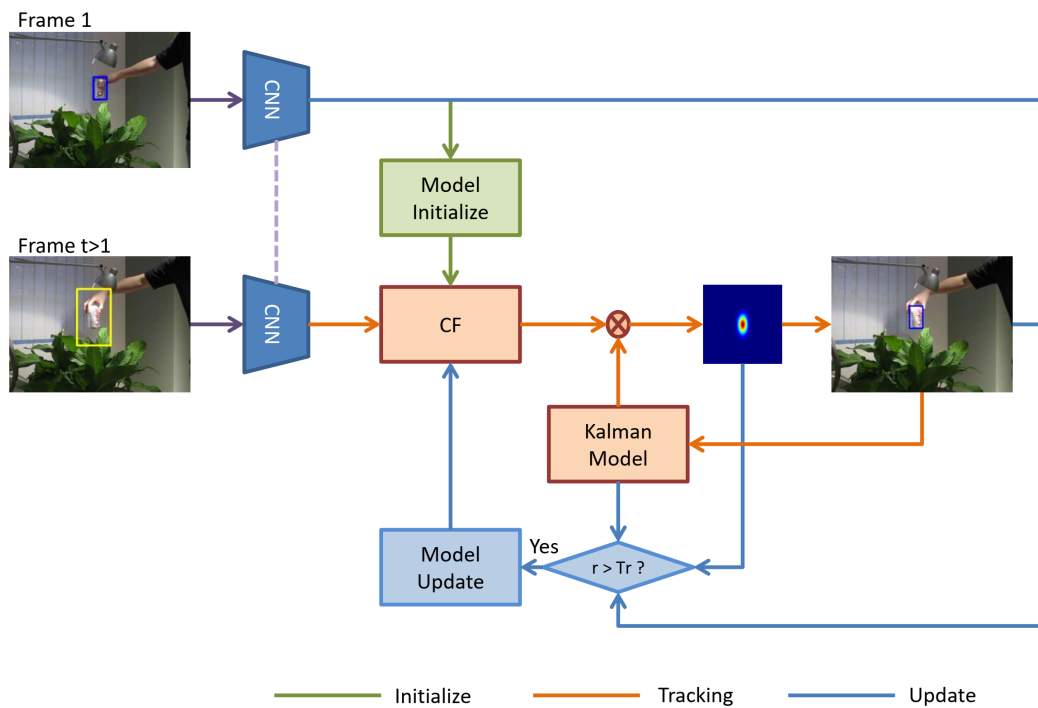


Figure 2. Pipeline of our proposed method. Firstly, the correlation filter is initialized with target information in the first frame. Secondly, the correlation filter (CF) and Kalman filter are used to estimate the target location in the subsequent frames, and then the reliability of the estimation is analyzed by reliability analysis module. Once the localization is reliable, the model is trained jointly by using the target information in the initial frame and the current reliable information. Finally, CF is updated. Besides, r is the reliability of the tracking result; T_r is the threshold to determine whether the tracking is reliable. In addition, we use CNN to extract image features.

3.1. Reliability Analysis

The response map represents the tracking result on the current frame. So we can calculate the reliability of the tracking process by analyzing the response map. Response reliability can be analyzed through two aspects: precision and stability as shown in Figure 3. Intuitively, a larger maximum response corresponds to a higher accurate location. The precision corresponds to the magnitude of response of the correlation filter. So the precision reliability is expressed as

$$\mu_l = \max(S_l), \quad (1)$$

where S_l represents the response map of frame l .

Stability reliability corresponds to the quality of filtering response. Peak Sidelobe Ratio (PSR) is mentioned in the MOSSE tracker as a criterion to measure detection process. PSR indicates the quality of filtering response and whether tracking drift occurs. It calculate the ratio of sub-peak to main peak to estimate the reliability of tracking process.

$$PSR = \frac{r_{sub}}{r_{main}}, \quad (2)$$

where r_{sub}, r_{main} represent sub-peak and main peak respectively.

However, this method has one problem, it can not deal with the problem of similar object interference. For example, when a new similar object appears, there may be a higher sub-peak around the main peak, resulting in a larger PSR value, but this does not indicate that the tracking fails, and the tracking result is still reliable. So we improve the stability reliability calculation as follows:

$$\rho_l = 1 - \min\left(\frac{r_{mean}}{r_{max}}, 0.6\right), \quad (3)$$

where r_{max} and r_{mean} are the main peak and the mean value of response map respectively. Threshold 0.6 is also used to mitigate the penalty when similar objects appear in the search image.

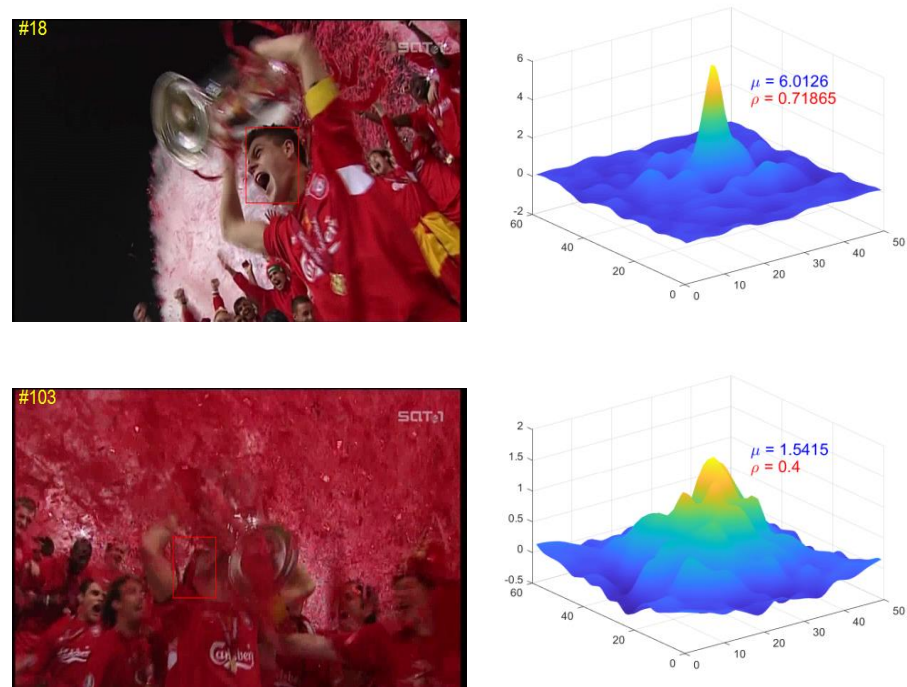


Figure 3. The images in first column are frames of sequence Soccer, and corresponding precision reliability and stability reliability are placed in the second column. Obviously, the reliability of target image in first row is higher than that in second row.

From Equations (1) and (3), we can calculate the final tracking process reliability

$$r_l = \min\left(\frac{\mu_l}{\mu_{max}} \cdot \rho_l, 1\right), \quad (4)$$

where μ_{max} is the maximum value of all response maps that have been tracked. When the target response reliability meets the condition of $r_l > T_r$, it indicates that the tracking process is reliable. T_r is a reliability threshold, which is set to 0.6 in this paper.

3.2. Trajectory Modeling and Kalman Filter

Most of the traditional tracking methods [7,8,11,15,24] only focus on the target detection in the current frame in the tracking process, and rarely model the temporal information of the target between frames. Visual tracking is based on video image sequence, so temporal information is very important for robust tracking, especially in challenge tracking scenes such as occlusion and the existence of similar distractors. In this paper, we use KF to model the motion of the object and get the trajectory information.

Kalman filtering (KF) is an algorithm that uses the state equation of linear system to estimate the system state through the input and output observation data of the system. Given the system parameters, initial values and measurement sequences, the KF can estimate the system state sequences iteratively. For the tracking tasks, because there is no control variable, we can first ignore the input, and the process noise and observation noise can be set as white noise. The label given in the initial frame is a bounding box (x, y, w, h) . In the motion model, we only consider the position information, so we can formulate the system state as a 4-dimensional vector (x_c, y_c, v_x, v_y) , where x_c, y_c represent the object center, v_x, v_y is the velocity in both directions. In this paper, we approximate the translation between frames as a constant velocity model.

$$\begin{cases} X_k = FX_{k-1} \\ Z_k = HX_k + V_k \end{cases} \quad (5)$$

where F is the state transition matrix, H is measurement matrix, V_k is measurement noise, and Z_k is the measurement.

The process of state estimation can be divided into two steps: prediction step and update step. We can use the following formulas to perform the prediction step:

$$\hat{X}_{k,k-1} = F\hat{X}_{k-1}, \quad (6)$$

$$P_{k,k-1} = FP_{k-1}F^T, \quad (7)$$

where \hat{X}_{k-1} is the optimal estimation of the previous state, P_{k-1} is the error covariance matrix of the previous optimal state estimation.

The formula for calculating the Kalman gain is as follows:

$$K_k = P_{k,k-1}H^T(H P_{k,k-1}H^T + R)^{-1}, \quad (8)$$

where R is the measurement error covariance matrix.

The measurement in this paper can be set as the output of CF. At last, the predicted values can then be updated:

$$\hat{X}_k = \hat{X}_{k,k-1} + K_k(Z_k - H\hat{X}_{k,k-1}), \quad (9)$$

$$P_k = (I - K_kH)P_{k,k-1}, \quad (10)$$

where \hat{X}_k is the posterior estimation of current state, I is the identity matrix, and P_k is the error covariance matrix of the current state estimation. Thus, we get the optimal state estimation of the current step through the motion model. The optimal estimation can be regarded as a refined update of CF tracking results in visual tracking task. It can be a powerful supplement to CF tracking method.

3.3. Reliable Online Training

The first frame contains the only absolutely reliable information of the target. Maintaining the first frame information in the tracker is very important for robust tracking. In this paper, we combine the first frame target information with the current target information to train a reliable correlation filter.

Firstly, we review the traditional CF tracking methods. The principle of CF method is extremely simple while tracking at very high frame rate and maintaining high tracking performance. The core advantages of this method lie in two points: (1) A Large number of approximate samples are obtained by intensive sampling of the original signal through cyclic shift. (2) In the process of training and detection, correlation operations are converted into frequency domain, to simplify the calculation greatly. The CF methods reformulate the tracking process as a ridge regression problem, train the filter through the samples and labels, and then use the filter to locate the target in search patch and update the filter on newly located object. The objective function of ridge regression can be expressed as follows:

$$\min_w \|Xw - y\|^2 + \lambda_1 \|w\|^2, \quad (11)$$

where sample matrix X contains the data vector x and all its cyclic shift versions as row vector, w is the correlation filter to be learned, y is the Gaussian shape regression response corresponding to all samples, λ_1 is a regularization parameter to prevent over-fitting of the model.

Traditional CF trackers use linear interpolation to update the filter, which makes the reliable initial target information decrease exponentially. These methods are effective for tracking under simple situation. For challenging scenarios, noise information will distort the learned filter, which decline the tracking performance or even lead to tracking drift. We use reliable object information in initial frame and current object information to enhance target foreground information and reduce the impact of noise.

Suppose that each target image X_l has M -dimensional features $X_l^i, i = 1, \dots, M$, the corresponding filter for each feature channel is $w_i, i = 1, \dots, M$. We reformulate Formula (11) to

$$\min_w \sum_{l=1}^2 \beta_l \left\| \sum_{i=1}^M X_l^i w_i - y \right\|^2 + \lambda_1 \sum_{i=1}^M \|w_i\|^2, \quad (12)$$

where $\beta_l, l = 1, 2$ are the weights for the templates.

The summation formula can also be written in vector form

$$\min_w \|\bar{X}\bar{w} - \bar{y}\|^2 + \lambda_1 \|\bar{w}\|^2, \quad (13)$$

$$\bar{X} = \begin{bmatrix} \sqrt{\beta_1} X_1^1 & \cdots & \sqrt{\beta_1} X_1^M \\ \sqrt{\beta_2} X_2^1 & \cdots & \sqrt{\beta_2} X_2^M \end{bmatrix}, \bar{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix}, \bar{y} = \begin{bmatrix} \sqrt{\beta_1} y \\ \sqrt{\beta_2} y \end{bmatrix}.$$

Formula (13) can be solved by setting the gradient of the objective function to zero

$$\bar{w} = (\bar{X}^T \bar{X} + \lambda_1 I)^{-1} \bar{X}^T \bar{y}. \quad (14)$$

According to the properties of cyclic matrix, fast calculation is carried out in frequency domain. The solution in frequency domain is as follows

$$\hat{w} = \begin{bmatrix} D_{11} + \lambda_1 & \cdots & D_{1M} \\ \vdots & \ddots & \vdots \\ D_{M1} & \cdots & D_{MM} + \lambda_1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^2 \beta_i (X_l^1)^* \odot \hat{y} \\ \vdots \\ \sum_{i=1}^2 \beta_i (X_l^M)^* \odot \hat{y} \end{bmatrix}, \quad (15)$$

$$D_{ji} = \sum_{l=1}^2 \beta_l \text{diag}((\hat{X}_l^j)^* \odot \hat{X}_l^i), j, i = 1, \dots, M,$$

where \odot represents element-wise multiplication, $*$ denotes complex conjugate. The variable with hat represents its corresponding Fourier transform.

For multi-channel, the primal domain detection needs to use the corresponding filter to detect in each channel of search image Z , and finally all the channel detection results are integrated

$$\hat{f}(Z) = \sum_{i=1}^M \hat{z}_i \odot \hat{w}_i. \quad (16)$$

The solution in dual space is

$$\bar{\alpha} = (\bar{X}\bar{X}^T + \lambda_1 I)^{-1} \hat{y}. \quad (17)$$

Then, according to the properties of cyclic matrix, it is converted to frequency domain for calculation

$$\hat{\alpha} = \begin{bmatrix} D_{11} + \lambda_1 & D_{12} \\ D_{21} & D_{22} + \lambda_1 \end{bmatrix}^{-1} \begin{bmatrix} \sqrt{\beta_1} \hat{y} \\ \sqrt{\beta_2} \hat{y} \end{bmatrix}, \quad (18)$$

where

$$D_{jl} = \sum_{i=1}^M \sqrt{\beta_j \beta_l} \text{diag}(\hat{X}_j^i \odot (\hat{X}_l^i)^*), j, l = 1, 2.$$

At last, the detection formula is

$$\hat{f}(Z) = \sum_{l=1}^2 \sum_{i=1}^M \hat{z}^i \odot (\hat{X}_l^i)^* \odot \hat{\alpha}_l. \quad (19)$$

3.4. Filter Update

Most traditional CF trackers adopt strict frame-by-frame update strategy. However, the target information between adjacent frames changes little and has much redundant information, which not only slows down the tracking speed, but also makes the tracking performance degraded when facing complex tracking environment. Many researchers also proposed improved method to update every N frames, but it still exist the problem of inaccurate object information. In our method, we adopt the strategy of sparse updating and reliability analysis of target information. Therefore, we can get more robust and accurate updated filters. To obtain a better performance and avoid drastic change of model, we use a moving average method to update correlation filter.

$$\hat{w}_i^t = (1 - \delta) \hat{w}_i^{t-1} + \delta \hat{w}_i^{new}, \quad (20)$$

$$\hat{\alpha}_l^t = (1 - \eta) \hat{\alpha}_l^{t-1} + \eta \hat{\alpha}_l^{new}, \quad (21)$$

where δ, η are the corresponding learning rates.

3.5. Tracking Details

Deep features are extracted by VGG-Net-19 network [32] which removes all the full-connection layers. The network is pre-trained on the ImageNet [33] ILSVRC dataset to perform classification tasks, and the deep features extracted by VGG have also been used in many other fields [11,34]. Instead of just using the output of the last layer of the network, we use the output of 3-4, 4-4 and 5-4 layers. This is because the high-level features tend to be semantic, with high stability but low resolution, which is conducive to improve the robustness of the tracking process. The low-level features are more texture oriented, with low stability but high resolution, which is conducive to improve the accuracy of localization process. As shown in Figure 4, we calculate the response maps on the features of three layers. Finally, we fuse the three response maps to get the final result,

$$S_f = \sum_{l=3}^5 w_l S_l, \quad (22)$$

where S_f is the fused result, $S_l, l = 3, 4, 5$ represent response maps of different layer features. w_l is the weight for fusing.

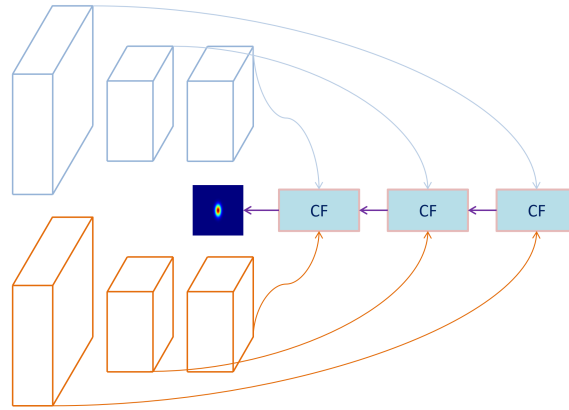


Figure 4. Details of the fusion process of response maps. The blue block represents the correlation filter. First, the three stage features are fed into three correlation filters to obtain three response maps, and then all the results are fused to compute the final location estimation.

In order to improve the accuracy and stability of the tracking process and make the filter more suitable for future tracking targets, we use the reliability values of current frame to calculate the weight

$$\beta_1 = \gamma_l, \quad \beta_2 = 1 - \beta_1. \quad (23)$$

The detailed tracking method is shown in Algorithm 1.

Algorithm 1 Robust Visual Tracking with Reliable Object Information and Kalman Filter

Input: Initial target position p_0

Output: Estimated target position p_t and updated correlation filters

- 1: Initialize the filters according to p_0 , and save object features X_0
 - 2: **repeat**
 - 3: According to the p_{t-1} and correlation filters, calculate the \bar{p}_t in frame t ;
 - 4: Taking the computed \bar{p}_t as observation, estimate the target position \hat{p}_t by Kalman filter;
 - 5: Fuse the results of two modules, and obtain p_t ;
 - 6: According to the fusion confidence map, analyse the reliability of the tracking process;
 - 7: **if** *reliability* > *Threshold* **then**
 - 8: Send X_0, X_t into the online training module, and update the filters;
 - 9: **else**
 - 10: Continue;
 - 11: **end if**
 - 12: **until** The last frame of the sequences
-

4. Experiments

In this section, we firstly demonstrate the effectiveness of this method with ablation experiments. Then, we compare our proposed method with state-of-the-art trackers on dataset OTB-2013 [35], OTB-2015 [36], VOT2016 [37], and VOT2018 [38].

4.1. Evaluation Criteria and Parameter Setting

On OTB dataset, we use OPE criterion [5,37] to evaluate all trackers, which including two metrics: precision and success rate. Precision is the Euclidean distance between the center position of estimated result box and ground truth bounding box. Twenty pixel distance threshold is usually used to compare the performance of each tracker. Success rate is a measure of the overlapping area of two boxes

$$IOU = \frac{\text{area}(B_T \cap B_G)}{\text{area}(B_T \cup B_G)}, \quad (24)$$

where B_T, B_G are the estimation and ground truth respectively, \cap, \cup denote the intersection area and union of two boxes. When the overlap area exceeds a certain threshold, such as $IOU \geq 0.5$, we assume that the tracking in this frame is successful. The success rate can be obtained by dividing the number of frames successfully tracked by the total number of frames. Area Under Curve (AUC) value is usually used to ranking the trackers in success plot.

In the VOT protocol, the trackers need to be reinitialized when tracking fails. Trackers performance is measured by accuracy and robustness, which correspond to the bounding box average overlap during successful tracking and failure rate, respectively. Expected Average Overlap (EAO) is used to evaluate the overall tracker performance. Please refer to VOT2016 [37] for details.

We have implemented the proposed method in MATLAB, in which the implementation of convolution neural network is based on MatConvNet toolbox [39]. All trackers run on the same computer equipped with Intel Core i7-8700 CPU, 16GB RAM and a NVIDIA GTX 1080 GPU.

4.2. Ablation Experiments

In this section, we conduct ablation experiments on OTB dataset, and analyze the effectiveness of each module proposed in this paper. We use DCF as the baseline tracker, but the difference is that we use convolution network to extract features. In order to test the performance of different components, we build three different trackers using baseline tracker and each component: (1) *Baseline + RA* is constructed by baseline and reliability analysis module, (2) *Baseline + KF* is constructed by baseline and Kalman Filter, (3) *Baseline + OT* indicates that the updated filter is trained by target information of the first frame and the current frame.

The overall experimental results are shown in Figure 5. The left figure shows the experimental results of the accuracy measurement. The number in the legend is the tracking precision when the distance error threshold is 20. The right figure shows the total success rate of each tracker. The number in legend is the AUC (area under curve). In precision plots, *Baseline + All* obtains the optimal performance of 85.3%, 84.6% on OTB2013, OTB2015, respectively. Compared with the other four constructed trackers, the precision performance gains on OTB2013 are 2.3%, 3.2%, 4.3% and 5.9%, and those on OTB2015 are 2.1%, 3.0%, 4.3% and 6.1%, respectively. Similarly, *Baseline + All* obtains precision scores of 63.2%, 62.1% on OTB2013, OTB2015 in success plots. Compared with the other four trackers, the success gains on OTB2013 are 1.6%, 2.9%, 4.5% and 6.7%, and those on OTB2015 are 1.5%, 2.5%, 4.3% and 5.9%, respectively. We can see that KF has the least improvement in the performance of the benchmark tracker among the three modules. This is because the accuracy of the benchmark tracker is low, which makes the measurement error in the KF process larger and leads to suboptimal estimation results, which makes a single KF module improve the performance of the benchmark tracker less than the other three

modules. The RA module contributes the most to the performance gain of the baseline tracker, this shows that reliable object information is very important for robust tracking process. OT module also plays an important role in improving the performance of the benchmark tracker. This is because the module always keeps a certain initial frame target information in the model. For occlusion, long sequence and other scenes, it can effectively keep the reliable information of the target and avoid tracking drift and failure.

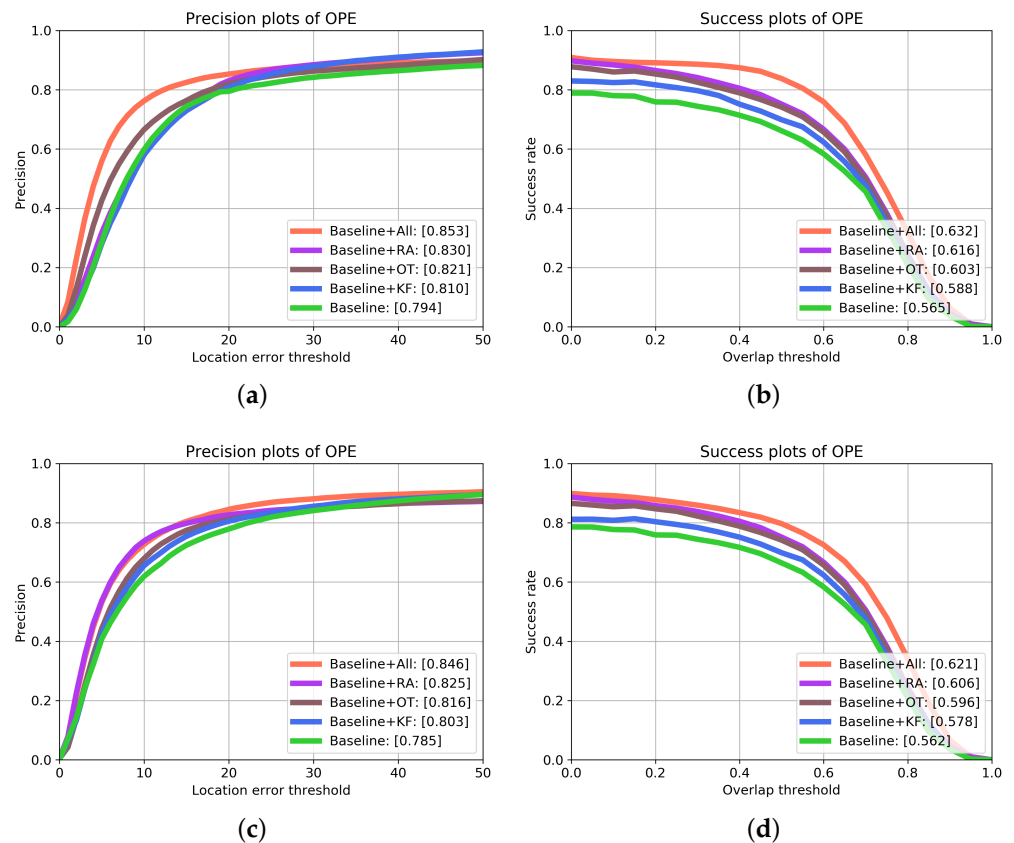


Figure 5. Overall results of ablation experiments by using one-pass-evaluation on the OTB-100 dataset. The upper and lower pairs are the results on OTB2013 and OTB2015 respectively.

The OTB dataset is manually tagged with 11 different attributes, which represents the challenging aspects. These attributes include—illumination Variation, Occlusion, Fast Motion, Background Clutters, Out-of-Plane Rotation, Deformation, In-Plane Rotation, Low Resolution, Scale Variation, Motion Blur, Out-of-View. These subsets based on attributes play an important role in evaluating tracker performance and further improvement. Tables 1 and 2 show the precision scores and AUC scores of each tracker on the 11 attribute based subset, respectively. We can see that *Baseline + All* has an absolute advantage over other trackers in all attribute subsets. Our proposed framework has improved the performance of the baseline tracker greatly, and the performance gain of the baseline tracker based on each module is consistent with the total result. This further confirms that reliable target information is the most important for the tracking process, and KF also provides important supplementary information for robust tracking.

Table 1. Precision scores of ablation experiments on 11 attributes of OTB100.

	Baseline + All	Baseline + RA	Baseline + OT	Baseline + KF	Baseline
Illumination Variation	0.848	0.823	0.813	0.796	0.786
Occlusion	0.831	0.801	0.796	0.775	0.771
Fast Motion	0.763	0.742	0.741	0.726	0.706
Background Clutters	0.856	0.836	0.824	0.819	0.786
Out-of-Plane Rotation	0.828	0.811	0.795	0.791	0.782
Deformation	0.852	0.831	0.825	0.814	0.805
In-Plane Rotation	0.830	0.822	0.814	0.807	0.789
Low Resolution	0.798	0.776	0.765	0.758	0.737
Scale Variation	0.817	0.801	0.796	0.785	0.754
Motion Blur	0.788	0.765	0.753	0.750	0.736
Out-of-View	0.721	0.709	0.691	0.679	0.667

Table 2. Area under curve (AUC) scores of ablation experiments on 11 attributes of OTB100.

	Baseline + All	Baseline + RA	Baseline + OT	Baseline + KF	Baseline
Illumination Variation	0.603	0.582	0.579	0.564	0.556
Occlusion	0.598	0.581	0.572	0.552	0.539
Fast Motion	0.573	0.558	0.546	0.530	0.524
Background Clutters	0.628	0.612	0.609	0.597	0.583
Out-of-Plane Rotation	0.615	0.592	0.583	0.572	0.551
Deformation	0.606	0.586	0.574	0.559	0.531
In-Plane Rotation	0.624	0.608	0.582	0.583	0.568
Low Resolution	0.542	0.521	0.516	0.503	0.496
Scale Variation	0.581	0.571	0.559	0.542	0.532
Motion Blur	0.599	0.583	0.577	0.552	0.548
Out-of-View	0.532	0.519	0.503	0.498	0.482

4.3. Comparison with Other Trackers

In order to analyze and evaluate the proposed tracker more comprehensively, we compare it with other trackers on OTB and VOT datasets.

OTB Dataset. We compare our tracker with 18 latest methods: TLD [40], CSK [13], MOSSE [12], Struck [41], KCF [7], DSST [15], CFNet [11], Staple [42], SiamFC [24], SiamDCF [43], SiamTri [44], SRDCF [8], DLSSVM [45], CNN-SVM [46], ACFN [30], SRDCFad [47], DeepSRDCF [16], TRACA [48]. We also carried out experiments on OTB2013 and OTB2015, respectively.

Figure 6 shows the tracking performance of all trackers on benchmark OTB2013. Our tracker achieves the second-best performance in distance precision score of 86.3%, but the AUC score of 65.1% outperforms all 18 other trackers. The best tracker TRACA outperforms our tracker by 1.9% in distance precision, but its AUC performance is 0.8% lower than ours. Figure 7 illustrates the tracking performance of all trackers on OTB2015 dataset. We can see that our tracker's AUC and DP scores are 85.6% and 64.5% respectively, which makes our tracker completely outperforms all other trackers in two indicators. The AUC and DP scores of the best performance tracker TRACA on OTB2013 decreased by 7.7% and 4.6% on OTB2015, respectively. The performance of ACFN on OTB2015 is also decreased by 5.2% and 2.9%. Different from many other trackers, the performance of our tracker on OTB2015 is slightly lower than that of OTB2013, which decreases by 0.7% and 0.6% respectively. This shows that our tracker can deal with complex tracking scene better and has high tracking robustness. Overall, the experiment results on two benchmarks demonstrate that our tracker performs well against other 18 tracker.

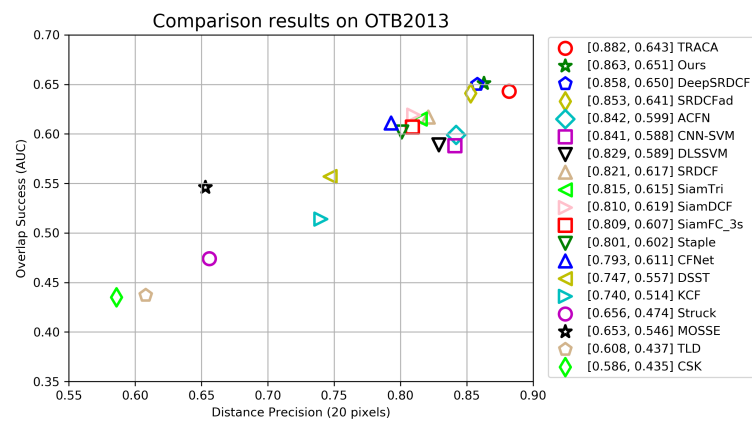


Figure 6. The precision and success scores of our tracker and other trackers on the OTB2013 dataset. The two columns of numbers in the legend represent the AUC score and the precision score at a threshold of 20 pixels. All trackers are sorted in the legend by precision scores.

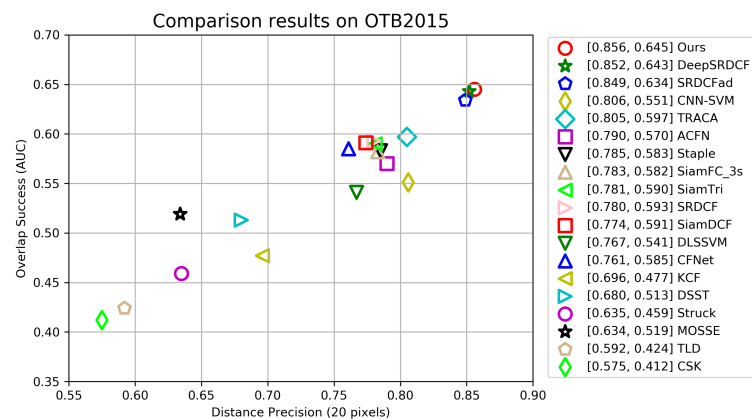


Figure 7. The precision and success scores of our tracker and other trackers on the OTB2015 dataset. The two columns of numbers in the legend represent the AUC score and the precision score at a threshold of 20 pixels. All trackers are sorted in the legend by precision scores.

VOT2016 Dataset. The VOT2016 dataset contains 60 short video sequences, and the accuracy (A), robustness (R) and expected average overlap (EAO) are three important criterion for evaluating trackers. In addition, EFO is often used to measure tracking speed. We compare our approach with 18 other state-of-the-art tracking algorithms on the VOT2016 benchmark. Figure 8 shows the EAO scores and rankings of all trackers on VOT2016. The best tracker is CCOT, with an EAO score of 0.331. Our tracker ranks second, with a performance slightly lower than that of CCOT, with an EAO score of 0.328. It is worth noting that the trackers above the horizontal line in the figure can be considered as state-of-the-art. Table 3 reports the detailed performance information about ours and several top trackers on VOT2016. Of all the trackers, our tracker ranked fourth in accuracy and first in robust. Although our tracker's accuracy score is inferior to the top three, it is only 0.5% lower than the best tracker. Our tracker ranks first in robust, which shows that reliable target information and motion information are very important to the robustness of the tracker. Our tracker can adapt to a variety of challenging tracking scenarios.

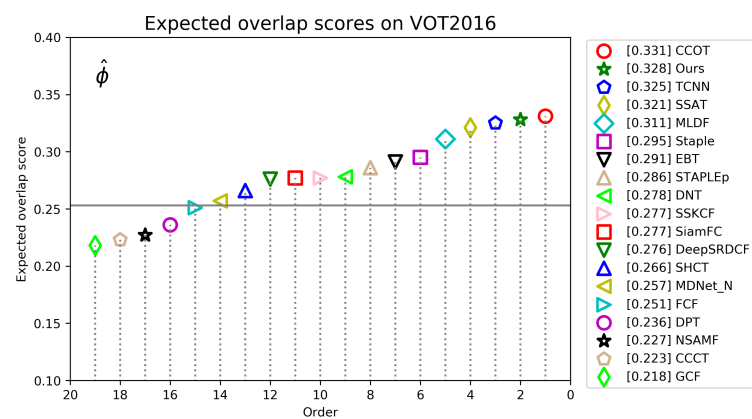


Figure 8. Expected average overlap scores ranking for compared trackers on the VOT2016 benchmark. The further to the right, the better the performance of the tracker.

Table 3. Detailed performance information about ours and several top tracker on VOT2016. Red, blue and green highlighted numbers indicate the 1st, 2nd and 3rd respectively.

	Ours	CCOT	TCNN	SSAT	MLDF	Staple	EBT	STAPLEp	DNT	SSKCF	SiamFC
EAO \uparrow	0.328	0.331	0.325	0.321	0.311	0.295	0.291	0.286	0.278	0.277	0.277
Accuracy \uparrow	0.552	0.539	0.554	0.577	0.490	0.544	0.465	0.557	0.515	0.547	0.549
Robust \downarrow	0.230	0.238	0.268	0.291	0.233	0.378	0.252	0.368	0.329	0.373	0.382
EFO \uparrow	10.16	0.507	1.049	0.475	1.483	11.14	3.011	44.77	1.127	29.15	5.444

VOT2017 Dataset. VOT2017 maintains 60 video sequences just like VOT2016. The difference is that VOT2017 removes 10 least challenging sequences from VOT2016, and adds 10 new sequences while keeping the overall attribute distribution unchanged. At the same time, it also re-calibrates the ground truth of all sequences. Figure 9 shows the EAO scores and rankings of all compared trackers on VOT2017. We can see that the best tracker is LSART with an EAO score of 0.323, while our tracker ranks third with an EAO score of 0.287. CCOT, the best tracker in VOT2016, has an EAO score of 0.267, which is 2.0% lower than our tracker. This is mainly due to the replacement of 10 new sequences, which makes the VOT2017 dataset more challenging than VOT2016, and our tracker has higher robustness in complex scenes, so our tracker performs better on VOT2017 than CCOT. Table 4 reports the detailed performance information about ours and 10 top trackers on VOT2017. We can see that our tracker ranked third in term of robustness with a score of 0.273, better than 0.318 of CCOT.

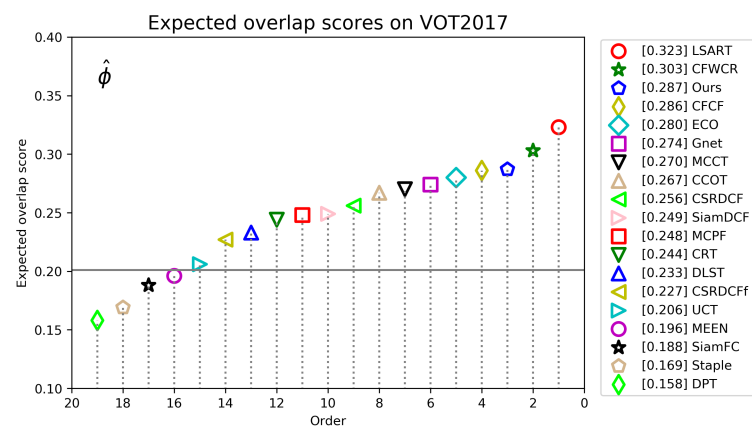


Figure 9. Expected average overlap scores ranking for compared trackers on the VOT2018 benchmark. The further to the right, the better the performance of the tracker.

Table 4. Detailed performance information about ours and several top tracker on vot2017. Red, blue and green highlighted numbers indicate the 1st, 2nd and 3rd respectively.

	Ours	LSART	CFWCR	CFCF	ECO	Gnet	MCCT	CCOT	CSRDCF	SiamDCF	MCPF
EAO \uparrow	0.287	0.323	0.303	0.286	0.280	0.274	0.270	0.267	0.256	0.249	0.248
A \uparrow	0.486	0.493	0.484	0.509	0.483	0.502	0.525	0.494	0.491	0.500	0.510
R \downarrow	0.273	0.218	0.267	0.281	0.276	0.276	0.323	0.318	0.356	0.473	0.427

4.4. Quantitative Results

In order to analyze the tracking performance more intuitively, we compare our tracker with 10 other trackers on several challenging video sequences on OTB datasets, and give the quantitative tracking results in Figure 10. We can see that our tracker can accurately locate the target under the influence of occlusion, long sequence, distractors and other factors. It shows that our tracker can reliably keep the target information, and obtain the motion information between frames through KF module, which makes it possible to deal with a variety of complex scenes. So our tracker achieves the best performance in these challenging sequences.



Figure 10. Quantitative tracking results of our tracker with other 10 trackers on OTB dataset. The video sequences are Couple, Doll, DragonBaby, Football1, Girl2 and Skating2_1 from top to bottom. The bottom illustration shows the colors for all trackers.

5. Conclusions

In this paper, We propose a robust visual tracking framework which mainly includes three modules: reliability analysis module, reliable online training and update module, and KF module. The reliability analysis module is mainly used to analyze the tracking process and identify whether the training update step can be carried out to prevent the introduction of noise information. The reliable online training update module is mainly to fuse the information of the first frame and the current frame to maintain the most reliable target information in the tracking process. KF module models the motion information between frames, which provides important supplementary information for our tracker. The proposed method improves the tracking performance of the tracker in complex scenes such as appearance change, tracking drift and occlusion. We validate the proposed framework on several benchmark datasets. Our tracker achieves the second and first AUC scores on OTB2013 and OTB2015, respectively. On VOT2016 and VOT2017 datasets, our tracker is also at the top. The tracking results show that our tracker achieves state-of-the-art performance. However, we observed that our tracker cannot deal with the deformation of objects very well. In future work, we will continue to optimize our tracker.

Author Contributions: H.C. constructed the tracking framework and performed the experiments and wrote the original manuscript. W.Z. and D.Y. analysed and interpreted the experiment results, and provided suggestions about the revision of this manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 62073266).

Data Availability Statement: Data of this research is available upon request via corresponding author.

Acknowledgments: Hang Chen would like to thank Ecco Guo for the support. The authors wish to thank the anonymous reviewers for their valuable suggestions.

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

References

1. Bouchrika, I.; Carter, J.N.; Nixon, M.S. Towards automated visual surveillance using gait for identity recognition and tracking across multiple non-intersecting cameras. *Multimed. Tools Appl.* **2016**, *75*, 1201–1221. [[CrossRef](#)]
2. Lien, J.; Olson, E.M.; Amihoud, P.M.; Poupyrev, I. RF-Based Micro-Motion Tracking for Gesture Tracking and Recognition. U.S. Patent No. 10,241,581, 26 March 2019.
3. Tokekar, P.; Isler, V.; Franchi, A. Multi-target visual tracking with aerial robots. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3067–3072.
4. Simon, M.; Amende, K.; Kraus, A.; Honer, J.; Samann, T.; Kaulbersch, H.; Milz, S.; Michael Gross, H. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Long Beach, CA, USA, 15–21 June 2019.
5. Smeulders, A.W.M.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual Tracking: An Experimental Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468. [[PubMed](#)]
6. Alper, Y.; Omar, J.; Mubarak, S. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, B1–B45.
7. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [[CrossRef](#)] [[PubMed](#)]
8. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 4310–4318.
9. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical convolutional features for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 3074–3082.
10. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488.
11. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.

12. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
13. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision (ECCV), Firenze, Italy, 7–13 October 2012; pp. 702–715.
14. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision (ECCV Workshops), Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
15. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference (BMVC), University of Nottingham, Nottingham, UK, 1–5 September 2014.
16. Danelljan, M.; Hager, G.; Shahbaz, Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV Workshops), Santiago, Chile, 11–18 December 2015; pp. 621–629.
17. Mueller, M.; Smith, N.; Ghanem, B. Context-aware correlation filter tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1387–1395.
18. Bibi, A.; Mueller, M.; Ghanem, B. Target response adaptation for correlation filter tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 419–433.
19. Xia, H.; Zhang, Y.; Yang, M.; Zhao, Y. Visual Tracking via Deep Feature Fusion and Correlation Filters. *Sensors* **2020**, *20*, 3370. [[CrossRef](#)] [[PubMed](#)]
20. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
21. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4293–4302.
22. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 FPS with deep regression networks. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.
23. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1420–1429.
24. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV Workshops), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
25. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4586–4595.
26. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4277–4286.
27. Kim, Y.; Shin, J.; Park, H.; Paik, J. Real-Time Visual Tracking with Variational Structure Attention Network. *Sensors* **2019**, *19*, 4904. [[CrossRef](#)] [[PubMed](#)]
28. Li, C.; Xing, Q.; Ma, Z. HKSiamFC: Visual-Tracking Framework Using Prior Information Provided by Staple and Kalman Filter. *Sensors* **2020**, *20*, 2137. [[CrossRef](#)] [[PubMed](#)]
29. Fan, H.; Ling, H. SANet: Structure-Aware Network for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Honolulu, HI, USA, 21–26 July 2017; pp. 2217–2224.
30. Choi, J.; Jin Chang, H.; Yun, S.; Fischer, T.; Demiris, Y.; Young Choi, J. Attentional correlation filter network for adaptive visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4807–4816.
31. Yang, T.; Chan, A.B. Learning Dynamic Memory Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 153–169.
32. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
33. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Fontainebleau Resort, Miami Beach, FL, USA, 20–25 June 2009; pp. 248–255.
34. Dong, H.; Ma, W.; Wu, Y.; Gong, M.; Jiao, L. Local Descriptor Learning for Change Detection in Synthetic Aperture Radar Images via Convolutional Neural Networks. *IEEE Access* **2018**, *7*, 15389–15403. [[CrossRef](#)]
35. Wu, Y.; Lim, J.; Yang, M. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 25–27 June 2013; pp. 2411–2418.
36. Wu, Y.; Lim, J.; Yang, M. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [[CrossRef](#)] [[PubMed](#)]
37. Lukežić, A.; Vojtír, T.; Kristan, M.; et al. The visual object tracking vot2016 challenge results. In Proceedings of the European Conference on Computer Vision (ECCV Workshops), Amsterdam, The Netherlands, 8–16 October 2016; pp. 777–823.
38. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Zajc, L.C.; Vojtír, T.; Bhat, G.; Lukežić, A.; Eldesokey, A.; et al. The sixth visual object tracking vot2018 challenge results. In Proceedings of the European Conference on Computer Vision Workshops – (ECCV Workshops), Munich, Germany, 8–14 September 2018; pp. 3–53.

39. Vedaldi, A.; Lenc, K. MatConvNet: Convolutional Neural Networks for MATLAB. In Proceedings of the 23rd ACM international conference on Multimedia, Brisbane, Australia, October 2015; pp. 689–692.
40. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [[CrossRef](#)] [[PubMed](#)]
41. Hare, S.; Saffari, A.; Torr, P.H.S. Struck: Structured output tracking with kernels. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 263–270.
42. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1401–1409.
43. Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; Hu, W. DCFNet: Discriminant correlation filters network for visual tracking. *arXiv* **2017**, arXiv:1704.04057.
44. Dong, X.; Shen, J. Triplet loss in siamese network for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 459–474.
45. Ning, J.; Yang, J.; Jiang, S.; Zhang, L.; Yang, M.-H. Object tracking via dual linear structured svm and explicit feature map. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4266–4274.
46. Hong, S.; You, T.; Kwak, S.; Han, B. Online tracking by learning discriminative saliency map with convolutional neural network. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 597–606.
47. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1430–1438.
48. Choi, J.; Jin Chang, H.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Young Choi, J. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 479–488.