



## Article

# Learning Local–Global Multiple Correlation Filters for Robust Visual Tracking with Kalman Filter Redetection

Jianming Zhang , Yang Liu, Hehua Liu and Jin Wang \* 

Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation,  
School of Computer and Communication Engineering, Changsha University of Science and Technology,  
Changsha 410114, China; jmzhang@csust.edu.cn (J.Z.); yang\_liu@stu.csust.edu.cn (Y.L.);  
lhh@stu.csust.edu.cn (H.L.)

\* Correspondence: jinwang@csust.edu.cn

**Abstract:** Visual object tracking is a significant technology for camera-based sensor networks applications. Multilayer convolutional features comprehensively used in correlation filter (CF)-based tracking algorithms have achieved excellent performance. However, there are tracking failures in some challenging situations because ordinary features are not able to well represent the object appearance variations and the correlation filters are updated irrationally. In this paper, we propose a local–global multiple correlation filters (LGCF) tracking algorithm for edge computing systems capturing moving targets, such as vehicles and pedestrians. First, we construct a global correlation filter model with deep convolutional features, and choose horizontal or vertical division according to the aspect ratio to build two local filters with hand-crafted features. Then, we propose a local–global collaborative strategy to exchange information between local and global correlation filters. This strategy can avoid the wrong learning of the object appearance model. Finally, we propose a time-space peak to sidelobe ratio (TSPSR) to evaluate the stability of the current CF. When the estimated results of the current CF are not reliable, the Kalman filter redetection (KFR) model would be enabled to recapture the object. The experimental results show that our presented algorithm achieves better performances on OTB-2013 and OTB-2015 compared with the other latest 12 tracking algorithms. Moreover, our algorithm handles various challenges in object tracking well.

**Keywords:** object tracking; correlation filter; convolutional neural networks; local–global collaborative strategy; Kalman filter



**Citation:** Zhang, J.; Liu, Y.; Liu, H.; Wang, J. Learning Local–Global Multiple Correlation Filters for Robust Visual Tracking with Kalman Filter Redetection. *Sensors* **2021**, *21*, 1129. <https://doi.org/10.3390/s21041129>

Academic Editor: SangMin Yoon

Received: 6 January 2021

Accepted: 1 February 2021

Published: 5 February 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

With the development of artificial intelligence, sensor networks nodes equipped with more sophisticated sensing units such as cameras have become ubiquitous in cities. Camera-based sensor networks have been widely used in intelligent transportation and smart cities. Because big video data are generated by cameras, the video analysis of smart nodes or edge servers needs to be energy efficient and intelligent. Visual object tracking is one of the fundamental problems in various application fields of computer vision, such as autonomous driving, precise navigation, video surveillance, and human–computer interaction. However, because of partial or complete occlusion, fast motion, lighting changes, scale changes, and similar objects and complex backgrounds, high-precision and robust visual target tracking is still a difficult task.

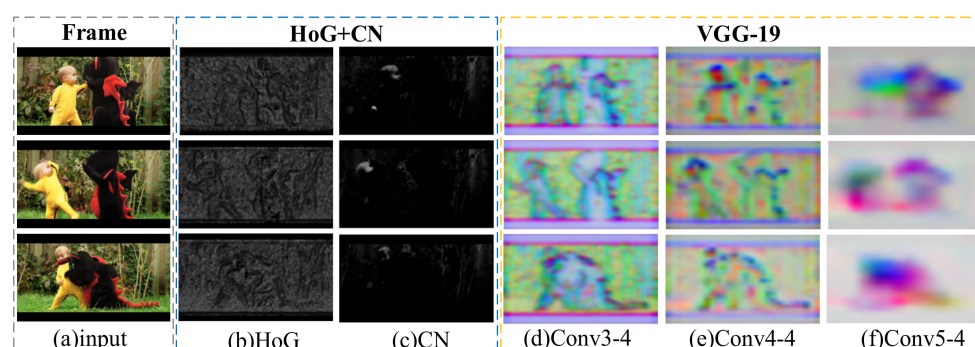
Building a target appearance model with strong representation capability is one of the keys to achieving the high precision and robustness of object tracking. According to the different categories of target appearance representation models used, the methods of visual object tracking can be divided into two types: generative methods [1,2] and discriminative methods [3,4]. The generative methods extract the target features for learning the appearance model representing the target at first, and then search the image area for pattern matching. The discriminative methods use the positive and negative samples collected from

each frame to train a binary classifier that distinguishes the target from the background; the candidate sample with the highest confidence in the current frame being selected as the tracking result. In recent years, discriminative correlation filter (DCF)-based trackers [5–7] have gained more and more attention from researchers due to their excellent performance. The DCF-based trackers can convert all samples into a diagonal matrix and transform the calculation into the frequency domain by Fourier transform, which greatly improves the speed of obtaining the solution to the correlation filter. The extraction method of the target appearance features mostly determines the performance of the DCF-based tracking algorithm. In order to train high-quality CF, it is necessary for the tracking algorithm to select an appropriate descriptor or a combination of multiple feature descriptors including a histogram of oriented gradient features (HoG) [8], color names (CN) features [9], Point of Interest features [10], Haar-like rectangular features [11], superpixel features [12], etc. This work uses a local–global multiple correlation filters model constructed by convolutional features and hand-crafted features to improve tracking performance.

### 1.1. Motivation

As a result of the strong discriminative capabilities of the convolutional features extracted by deep convolutional neural networks (CNN), they have been widely used in correlation filter tracking algorithms [13–16]. The features in the earlier layers of CNN retain high spatial resolution information that can be used for precise localization for targets, while rich semantic information contained in the features in the higher layers can be used to deal with the dramatic appearance changes for the target. General combination of low-level and high-level features can greatly improve the performance of the tracker, but there are still some limitations.

Deep features are used to represent targets in the CF-based trackers [17–20]. Compared with deep features, hand-crafted features are more robust to occlusion and shape changes. We visualize the above features as shown in Figure 1. It is important to improve the accuracy and robustness of the tracker to find a reasonable combination of multiple features. Moreover, due to the interference of factors such as occlusions and illumination changes during the tracking process, errors will gradually accumulate through the online update of CF, which will cause model drift and loss. Trackers with a single model can only learn one kind of appearance model, therefore the targets are prone to be lost. One recent approach [21] proposed a part-based multiple correlation filter tracking method that relies on the cooperation between a global filter and several part filters. However, this method uses only hand-crafted features, which makes it difficult to achieve excellent results. Thus, it is imperative to construct a reasonable multiple correlation filters cooperation mechanism.



**Figure 1.** Visualization of multi-level features. (a) Three challenging frames on the DragonBaby sequence. (b) Histogram of oriented gradient (HoG). (c) Color Name. (d–f) are high-level features extracted from Conv3-4, Conv4-4, and Conv5-4 of VGGNet-19.

These CF-based trackers have achieved rather good performance in visual object tracking. However, because they can only use the maximum value of the response map as the predicted result for candidate target position, it may be unreliable when the target

is out of view or completely occluded. More specifically, the response map may have multiple peaks or a decrease in maximum value when it comes to target occlusion or out of view objects. Therefore, an effective redetection mechanism is especially important in the tracking algorithm [22–24], when target occlusion and out of view occurs.

### 1.2. Contribution

In view of the above-mentioned two points, we put forward a local–global multiple correlation filter model and Kalman filter redetection mechanism to achieve accurate and robust object tracking. The main contributions are outlined below:

1. We propose an intelligent local–global multiple correlation filter (LGCF) learning model. Our global filter uses multilayer CNN features to capture the whole appearance of the target. At the same time, the object is divided into two equal sized parts to construct local filters, which use handcraft features to capture the partial appearance of the target. We choose a position estimation method from coarse to fine and use a combination of multiple features to achieve accurate object tracking.
2. We introduce an effective method where the target can be tracked accurately through Kalman filter redetection (KFR). We also propose time-space peak to sidelobe ratio (TSPSR) as a confidence value to measure the reliability of the current estimated position. Compared with CF with a different appearance model, we comprehensively use the time and motion information of the target instead of just the appearance information. If the tracking result of the correlation filter with the appearance model is unreliable, this method can make the target position regained to continue the tracking procedure.
3. Based on the local–global learning model, we propose a collaborative update strategy with multiple correlation filters. In the update phase, we divide the global filter into three cases according to the state of the local filter: normal update, slow update, and no update. This method can effectively prevent the collaborative filters from learning the wrong target appearance.
4. We have evaluated the performance of the proposed LGCF algorithm on the OTB-2013 [25] and OTB-2015 [26] datasets, and verified the effectiveness of each contribution. Experimental data show that this method can improve the tracking accuracy and success rate effectively. The precision and success rates on the OTB-2013 reached 90.6% and 64.9%, respectively. On OTB-2015, the accuracy of our tracker reached 86.2%, and the success rate reached 61.6%.

The remainder of this paper is organized as follows: Section 2 briefly introduces related work. The proposed tracking method is described in detail in Section 3. Experimental results are reported and analyzed in Section 4. We conclude our paper in Section 5.

## 2. Related Work

In this section, three types of tracking algorithms related to our algorithm are mainly introduced: tracking by correlation filters, tracking by deep learning, and tracking by Kalman filter (KF).

### 2.1. Tracking by CF

A tracking algorithm based on a correlation filter transforms the process of solving the filter into the frequency domain through a fast Fourier transform (FFT) to accelerate the calculation. Bolme et al. proposed the Minimum Output Sum of Squared Errors (MOSSE) [5] algorithm in 2010, which first introduced CF into visual tracking. The Kernelized Correlation Filter Tracker (KCF) uses CN and HoG features to compose multi-channel features, and uses kernels to map the calculation of low-dimensional feature space to linearly separable high-dimensional feature space [7]. Henriques et al. [8] propose Kernel Regularized Least Squares, which improves the speed of classifying samples in high-dimensional feature space. Bertinetto et al. [27] use statistical color histogram and HoG features to represent the target to train the CF. The fusion of these two features further improves tracking accuracy.

With the widespread application of convolutional neural networks [28,29], the hierarchical convolutional features tracker (HCF) [17] and Correlation Filters with Weighted Convolution Responses (CFWCR) [18] choose deep features extracted from VGGNet to train CF, and the tracking performance is further improved. Danelljan et al. [30] establish an interpolation model for training samples in [31], combine features in different resolutions to obtain feature maps with continuous spatial resolution, and use Gaussian mixture models to classify samples in the training set so that the similar samples are in one group. This method avoids overfitting between the samples in consequent frames. References [32,33] introduce multiscale estimation models. These two models have become commonly used in the process of object tracking. In order to solve the boundary effect caused by the periodic hypothesis, a spatial regularization term is introduced [6]. In the block object tracking [34], a probabilistic model is proposed to estimate the distribution of reliable patches under the sequential Monte Carlo framework. However, it is difficult for a filter trained by one type of feature to deal with the effects of multiple factors such as illumination and deformation at the same time. Therefore, we must find a reasonable way to combine multiple features. In our work, we comprehensively consider the invariance of hand-crafted features to geometry and illumination, as well as the fine-grained and semantic information of convolutional features. Therefore, the global filters are learned by multilayer convolutional features, while local filters are learned by the combination of HoG and CN features.

## 2.2. Tracking by Deep Learning

Deep convolutional neural networks require large-scale data for training, but this does not stop them from being widely used in object tracking due to their strong feature characterization capabilities. Some trackers directly apply deep features to the DCF framework, such as the Hedged deep tracker (HDT) [15]. Learning continuous convolution operators (CCOT) [31] and efficient convolution operators (ECO) [30] integrate the deep feature maps in different resolutions after interpolation. In addition, the Fully-convolutional Siamese network (SiamFC) [35] uses an end-to-end learning method and deep features to find a similarity between the target and the templates for object tracking. Dynamic Siamese (Dsiam) [36] can learn the appearance changes of the target online and use consequent video sequences for training to improve the tracking performance. The Siamese region proposal network (SiamRPN) [37] considers the tracking problem as a detection problem by utilizing a Regional Proposal Network (RPN) commonly used in object detection to obtain more accurate predictions of the tracking targets. A combined online tracking and segmentation method named SiamMask [38] introduces a binary mask branch to improve speed and accuracy in object tracking. Despite great success, trackers based on deep learning are still limited in many ways. On the one hand, a CNN model trained by a generic dataset such as ImageNet [39] is suitable for unspecified targets and is easily affected by background clutter. On the other hand, a tracker based on deep learning usually only learns the positive samples around the target and executes the strategy of not updating or slowly updating the model during the tracking process, which are easy to be influenced by occlusion or out of view. In this paper, we use multi-layer deep convolutional features extracted from VGGNet-19 [40] to train a global filter in the way of weighted combination. The global filter is used to achieve the coarse localization of the target. Then, HoG and CN features combined into 41-dimensional features are utilized to train our local filters. The local filters are used to achieve fine target localization and correct the coarse position estimation of global filters.

## 2.3. Tracking by KF

Kalman filter is an optimal linear recursive filter algorithm that can be easily implemented in a computer. It can effectively deal with problems relevant to filters under linear Gaussian. Multi-hierarchical Independent Correlation Filters for Visual Tracking (MFT) [41] uses KF to construct a motion estimation module to predict the trajectory of the tracking target. Reference [42] proposes Strong Tracking Marginalized Kalman Filter (STMKF) for



robust tracking with an edge Kalman filter. By using an attenuation factor in the Marginalized Kalman Filter (MKF), this reduces the impact of the previous filter on the current filter, considering the object in 3D space. Reference [43] analyzes the performance of extended KF using the azimuth to track single and multiple objects, respectively. Reference [44] compares the performance of the Kalman filter and particle filter for two-dimensional visual multi-object tracking in different environments. Reference [45] describes a method for detection and tracking on moving objects using KF. Aimed at cases where the object is occluded, this method can only rely on the state of the target in the previous frame to predict the target position in the current frame. However, the accuracy of the KF-based tracker is limited, and the mechanism of KF that needs to update every frame is likely to cause an accumulation of errors. Our method is similar to that in [45]. We use the tracking results of CF to replace target detection results. When the tracking target is occluded, the tracking result of the CF will drift or be lost. At this time, our KFR module will predict the target position in the current frame according to the status information of the target in the previous frame.

### 3. Our Approach

#### 3.1. Overview

In this paper, we propose a novel local–global tracking algorithm to achieve the combination of multiple features and introduce a redetection mechanism based on Kalman filter, which can achieve accurate object tracking. The overall framework of our proposed algorithm is shown in Figure 2. Algorithm 1 describes the main steps of our proposed LGCF tracker.

---

#### Algorithm 1 The Main Steps of Our Algorithm

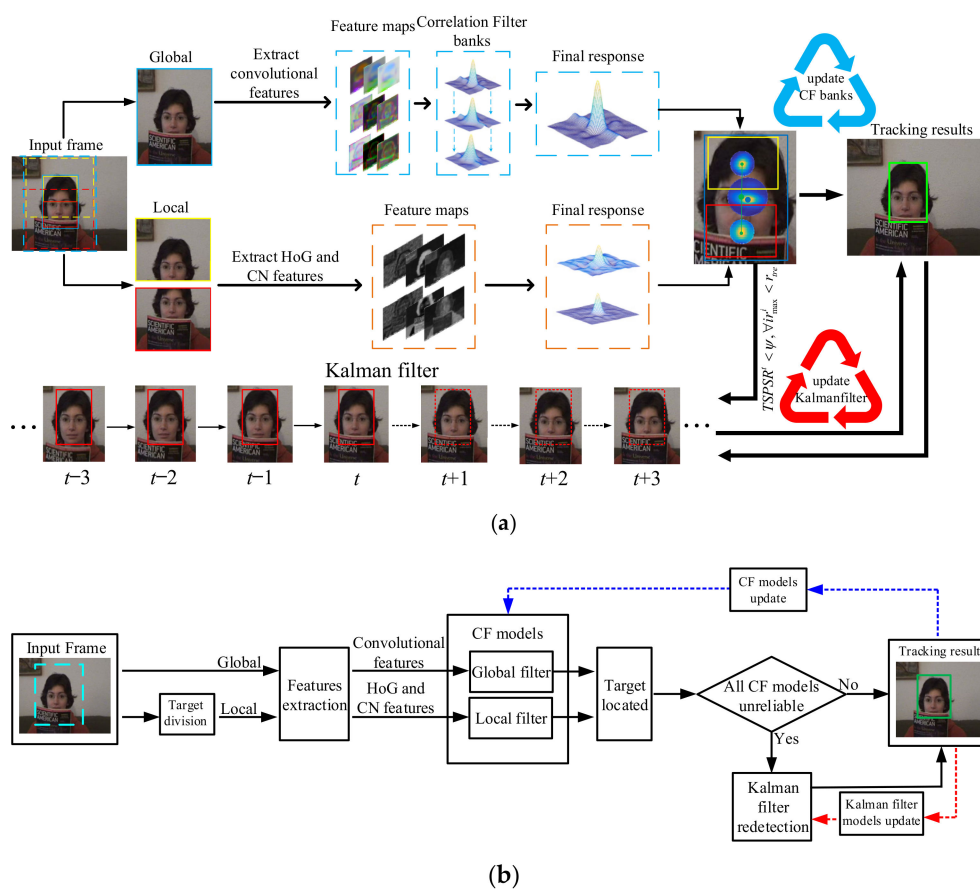
---

**Input:** Initial object position  $p_1$  in the first frame,

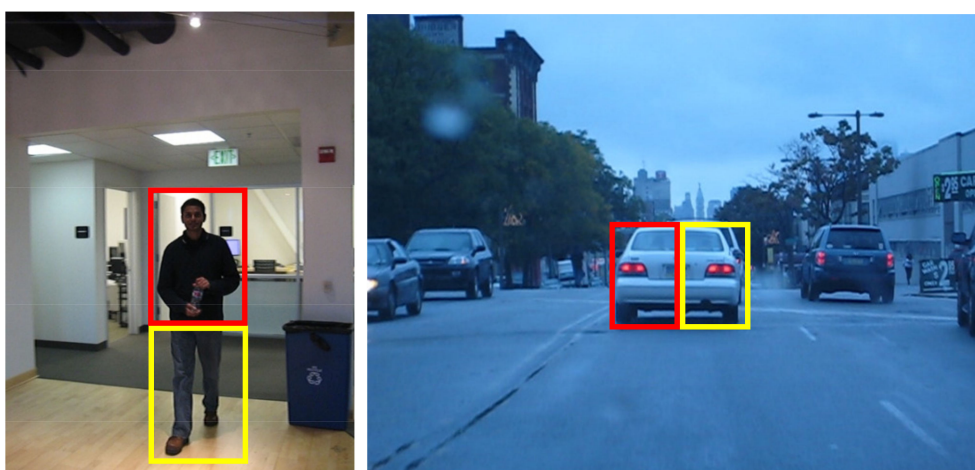
VGGNet-19 pretrained models;

**Output:** Estimated position  $p_t$ ; updated multiple correlation filters.

1. Divided target as Figure 3;
  2. Initiate global filter models using Equation (19);
  3. Initiate Kalman filter redetection models using Equations (22) and (23);
  4. **for**  $t = 2, 3, \dots$  **do**
  5. Exploit the VGGNet-19 to obtain multi-layer deep features;
  6. Compute the response maps of global filter using Equation (5);
  7. Compute the TSPSR using Equations (28) and (29);
  8. Find the final global target position  $pos_t^{(g)}$  using Equation (17);
  9. Compute the response maps of local filter using Equation (5);
  10. Find the final local target position  $pos_t^{(l)}$  using Equation (15);
  11. **if**  $TSPSR^t < \psi, \forall ir_{\max}^i < r_{tre}$  **then**
  12.     Obtain the target position of the current frame by Kalman filter using Equation (26);
  13.     Update Kalman filter redetection models using Equations (24)–(27);
  14. **else**
  15.     Obtain the target position of the current frame by multiple correlation filters using Equation (16);
  16. **end if**
  17. Update global filters using Equation (19);
  18. Update local filters using Equation (20);
  19. **end for**
-



**Figure 2.** System overview of our proposed tracking algorithm. We divide the target according to its aspect ratio, and then use a combination of local filters and global filters to achieve accurate object tracking. When the robustness scores of the CF are lower than a certain threshold, we enable the KFR mechanism to recapture the target. (a) Visualization of data flow in our proposed tracking algorithm. (b) The functional components and their connections in our proposed tracking algorithm. The blue dashed line indicates the collaborative update strategy of the correlation filter (CF), and the red dashed line indicates that the Kalman filter (KF) needs to be updated after Kalman filter redetection (KFR) is turned on.



**Figure 3.** The way to divide the object when building two local filters. This layout is selected according to aspect ratio of the target object. Sequences are from Human2 and BlurCar1, respectively.

First, our tracker consists of two parts: the global CF and the local CFs. The global CF is trained by multi-layer convolutional features extracted from VGGNet-19, while the local CF is trained by a combination of 31 dimensional HoG features and 11 dimensional

CN features. We directly solve the ridge regression problem in the linear space to learn our global filter, but map the ridge regression to the non-linear space through the kernel function to obtain the local filter. Our global filter and local filter can work together to make full use of the global overall information and local detailed information of the target. At the same time, the global and local filters can achieve information transfer and mutual correction.

Secondly, based on our local–global model, we propose a local–global filter collaborative update strategy. According to the different states of the local filter, we divide the target tracking states into three cases: no occlusion, partial occlusion, and complete occlusion. An optimal update strategy for the global filter is given for different tracking situations. This update strategy can prevent our filter from learning the fake target's appearance and error accumulation.

Finally, we propose TSPSR to measure the reliability of the current tracking results. Under normal circumstances, the tracking results calculated by the CF are fed to the KF for parameter update. In some challenging situations such as target loss, out-of-view rotation, or complete occlusion, the KF completely relies on the status and parameters of the target in the previous frame to predict the target position in the current frame. When the CF tracking results are unreliable, we will perform a resampling operation, and then use the KFR to estimate the current correct target position.

### 3.2. Global Filter

We use convolutional feature maps extracted from VGGNet-19 to build the global appearance of the target and train our global filter.  $x_l \in \mathbb{R}^{M \times N \times D}$  is denoted as the convolutional feature maps of size  $M \times N \times D$  from the  $l$ -th layer in VGGNet-19. The differences of  $M$ ,  $N$ , and  $D$  at the  $l$ -th level are not considered and  $x_l$  is regarded as the training sample. Circular samples  $\mathbf{x}_{m,n}, (m,n) \in \{0,1,\dots,M-1\} \times \{0,1,\dots,N-1\}$  are generated by circular matrices of  $x_l$ . Each circular sample  $\mathbf{x}_{m,n}$  corresponds to a two-dimensional Gaussian label function:

$$y(m,n) = e^{-\frac{(m-M/2)^2 + (n-N/2)^2}{2\sigma^2}}, \quad (1)$$

where  $M$  and  $N$  represent the width and height of the convolutional feature map, respectively;  $\sigma$  is the width of the kernel, and the correlation filters  $\mathbf{w}$  have the same size as  $\mathbf{x}_{m,n}$ . The goal is to find a function  $f(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$  that minimizes the squared error between the sample  $\mathbf{x}_{m,n}$  and the regression target  $y(m,n)$ :

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{m,n} \|\mathbf{w} \cdot \mathbf{x}_{m,n} - y(m,n)\| + \lambda \|\mathbf{w}\|_2^2, \quad (2)$$

where  $\lambda (\lambda > 0)$  is the regularization factor. The inner product is induced by a linear kernel, e.g.,  $\mathbf{w} \cdot \mathbf{x}_{m,n} = \sum_{d=1}^D \mathbf{w}_{m,n,d}^T \mathbf{x}_{m,n,d}$ . There are many channels of convolutional features; in order to simplify the calculation, we use a linear kernel in Hilbert space to derive  $f(\mathbf{x}_{m,n})$ :

$$f(\mathbf{x}_{m,n}) = \mathbf{w} \cdot \mathbf{x}_{m,n} = \sum_{d=1}^D \mathbf{w}_{n,m,d}^T \mathbf{x}_{m,n,d}. \quad (3)$$

Similar to the method in [46], using Fast Fourier transform (FFT) in each channel, the filter  $\mathbf{W}^d$  can be solved in the frequency domain. Each channel of the learned filter is represented in the frequency domain as follows:

$$\mathbf{W}^d = \frac{\mathbf{Y} \odot \mathbf{X}^{-d}}{\sum_{i=1}^D \mathbf{X}^i \odot \mathbf{X} + \lambda}, \quad (4)$$

where  $d \in \{1, 2, \dots, D\}$ ,  $\mathbf{Y} = \mathcal{F}(y)$ ,  $\mathbf{X}^d = \mathcal{F}(\mathbf{x}^d)$ , and  $\mathcal{F}(y)$  means Discrete Fourier transform (DFT). The bars above the variables represent complex conjugates. The operator  $\odot$  represents a Hadamard (element-wise) product. In the next frame, the convolutional features of the Region of Interest (ROI) are extracted, where the  $l$ -th layer is defined as  $\mathbf{z}_l \in \mathbb{R}^{M,N,D}$ . The layer  $l$  response map  $R_l$  is calculated as follows:

$$R_l = \mathcal{F}^{-1} \left( \sum_{d=1}^D \mathbf{W}^d \odot \mathbf{Z}^{-d} \right). \quad (5)$$

The operator  $\mathcal{F}(\cdot)^{-1}$  denote the inverse Fourier transform operation. We can obtain the estimated position of the target given by the  $l$ -th layer by searching the maximum value in the response map. We comprehensively consider the response maps generated by multiple feature layers. Specifically, we fuse the response maps of multiple convolutional layers by weights. The position of the maximum value of the weighted fusion response map is used to estimate the final position of global tracking:

$$\hat{R}(m, n) = \alpha_1 R_1(m, n) + \alpha_2 R_2(m, n) + \dots + \alpha_l R_l(m, n), \quad (6)$$

$$(x^{(g)}, y^{(g)}) = \arg_{m,n} \max \hat{R}(m, n), \quad (7)$$

$$pos^{(g)} = (x^{(g)}, y^{(g)}), \quad (8)$$

where  $\alpha_l$  is the weight of the  $l$ -th layer convolution feature response map.  $pos^{(g)}$  is the target position estimated by our global filter.

### 3.3. Local Filter

Our local filter does not rely on local variability between different targets and uses the variant of HoG [47] features and color name (CN) [48] features to capture the gradient and color information of the target, respectively. HoG features and CN features have low dimensionality, so we refer to the method in KCF [8] and use the Gaussian kernel function to map the ridge regression to a nonlinear space to construct two local trackers. For different parts of the target, each tracker is considered as a separate tracker. At the same time, they have constraints on each other and there are also constraints between partial filters and global filters. We refer to the block method in [21]. According to the aspect ratio of the target, we use two methods of horizontal block and vertical block to split the target, as the Figure 3 shows.

Using the kernel function to extend the ridge regression (Equation (2)) to the non-linear domain, we can obtain the closed-form solution of the best filter kernel version:

$$\alpha_i = (K + \lambda I)^{-1} \mathbf{y}_i. \quad (9)$$

Note that the subscript in Equation (9) indicates the  $i$ -th local filter.  $\alpha_i$  represents the form of the filter  $h_i$  in the dual domain.  $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$  is the kernel matrix,  $I$  is the identity matrix, and  $\mathbf{y}_i$  is a vector form of the desired output  $y_i$ .

For a kernel matrix with a cyclic structure, it is easy to construct a kernel solving Equation (2) without considering all the samples generated by the cyclic shifts. Given a kernel matrix  $K = C(\mathbf{k}^{\mathbf{x}}) \mathbf{k}^{\mathbf{x}}$  is the kernel correlation of  $\mathbf{x}$  with itself, where  $C(\cdot)$  represents the cyclic data matrix generated by concatenating all possible cyclic shifts. The closed-form solution can be written as:

$$\alpha_i = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(\mathbf{y}_i)}{\mathcal{F}(\mathbf{k}_i^{\mathbf{x}}) + \lambda} \right). \quad (10)$$

We use the HoG and CN features to construct our local target appearance representation and use the following Gaussian kernel to construct the kernel matrix:

$$\mathbf{k}^{\mathbf{x}\mathbf{x}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}\left(\sum_c \hat{\mathbf{x}}_c^* \odot \hat{\mathbf{x}}_c'\right)\right)\right), \quad (11)$$

where  $\hat{\mathbf{x}}$  is the Fourier transform of  $\mathbf{x}$ ,  $\hat{\mathbf{x}}^*$  is the complex conjugate of  $\hat{\mathbf{x}}$ , and  $\odot$  is element-wise. For the trained local filter  $\alpha_i$ , an image block of the same size as  $\mathbf{x}_i$  is cropped and its feature is  $\mathbf{z}$ . Confidence maps of the local filter and predicted target position are calculated as follows:

$$\hat{R}_i = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{k}_i^{\mathbf{x}\mathbf{z}}) \odot \mathcal{F}(\alpha_i)), \quad (12)$$

$$(x_i^{(l)}, y_i^{(l)}) = \underset{m,n}{\operatorname{argmax}} \hat{R}_i(m, n), \quad (13)$$

where  $\mathbf{k}_i^{\mathbf{x}\mathbf{z}}$  represents the kernel correlation of extracted feature  $\mathbf{z}_i$ , and the learned partial appearance model  $\alpha_i$ .  $\hat{R}_i$  represents the response map of the  $i$ -th local filter. We use the midpoint of the predicted position of the two local filters as the final position  $pos^{(l)}$  of local tracking:

$$(x^{(l)}, y^{(l)}) = \left(\frac{x_1^{(l)} + x_2^{(l)}}{2}, \frac{y_1^{(l)} + y_2^{(l)}}{2}\right), \quad (14)$$

$$pos^{(l)} = (x^{(l)}, y^{(l)}), \quad (15)$$

### 3.4. Local-Global Collaboration Model

We propose a global-to-local tracking algorithm, using the method in Figure 2 to build the target appearance model. Our global tracker is trained by convolutional features from the VGGNet-19 network, and the local tracker is trained by 31 dimensional HoG and 11 dimensional CN features. In the tracking process, the position and state of all filters are considered comprehensively to predict the final position of the target.

Specifically, whether the current local part of the target is occluded is determined by the maximum value of the local filter. When the maximum value of the local filter response  $r_{\max}^i$  is less than the threshold  $r_{tre}$ , we consider that the current local tracker results are not reliable. Considering the relationship between the local prediction position and the global prediction position, we set a reliable circle area around the global prediction position, the radius of which is related to the target scale, and the radius value is  $r = \sqrt{W^2 + H^2}$ . The Euclidean distance between the predicted position of the global tracking and the final position of the local tracking is calculated as  $d = \|pos^{(g)} - pos^{(l)}\|$ . To figure out whether the final position of the local tracking is within the reliable circle, that is, the relationship between  $r$  and  $d$ , the final tracking position is obtained as follows:

$$pos = \begin{cases} pos^{(g)} & \text{if } d < \theta \cdot r \\ (1 - \beta)pos^{(g)} + \beta pos^{(l)} & \text{if } d > \theta \cdot r, \exists ir_{\max}^i > r_{tre} \\ \text{KFR} & \text{if } d > \theta \cdot r, \forall ir_{\max}^i < r_{tre} \end{cases}, \quad (16)$$

where  $\Delta_{t-1}^{(l)}$  is the displacement vector of the center position of the two local correlation filters at time  $t$  relative to time  $t - 1$ .

For the global filter, the filter of the first layer can be updated by minimizing the output error of all training samples, which still includes solving the correlation filters of  $D$  channels. Online learning will consume a lot of time, so we update the numerator  $A_{t+1}^d$ , and denominator  $B_{t+1}^d$ , of the global filter  $H_{t+1}^d$ , respectively, as follows:

$$A_{t+1}^d = (1 - \eta)A_t^d + \eta_g G \odot \bar{F}_t^d, \quad (17)$$

$$B_{t+1}^d = (1 - \eta)B_t^d + \eta_g \sum_{i=1}^D F_t^i \odot \bar{F}_t^i, \quad (18)$$



$$H_{t+1}^d = \frac{A_t^d}{B_t^d + \lambda_g}, \quad (19)$$

where  $\eta_g$  is the update factor of the global filter. We use the maximum value of the local filter to determine whether the current target is occluded and whether the local–global learning model is updated is based on the state of the two local filters. We divide the tracking target status into the following three cases: no occlusion, partial occlusion, and complete occlusion. For three different cases, the global filter update method is as Table 1.

**Table 1.** Three different update states.

No	State of Tracker	State of Learning Rate	Value of Learning Rate	Condition in $t$ -th Frame
1	No occlusion	Normal update	$\eta_n$	$\forall i r_{\max}^i > r_{tre}$
2	Partial occlusion	Slow update	$\eta_p$	$\exists i r_{\max}^i > r_{tre}$
3	Complete occlusion	Not update	$\eta_c$	$\forall i r_{\max}^i < r_{tre}$

For local filters, we consider the state of each local filter separately. According to the maximum response values of the two local filters, the update method is selected as follows:

$$\alpha_i^t = \begin{cases} (1 - \eta_l) \alpha_i^{t-1} & \text{if } r_{\max}^i > r_{tre} \\ \alpha_i^{t-1} & \text{otherwise} \end{cases}, \quad (20)$$

where  $\eta_l$  is the update factor of the local filter. In the next section, we will discuss our proposed TSPSR index and how we recapture targets using KF when the local–global multiple CFs tracking model fails to track the target.

### 3.5. Kalman Filter Redetection

Most CF-based trackers and KCF trackers only consider appearance change, fail to utilize the motion information of the target, and rely too much on the maximum response value to determine the target position. When the target is partially or completely occluded or is in situations such as fast motion and complex background, the tracker may lose the target. Therefore, we propose a redetection module based on KF for the local–global tracker. KF is an algorithm that makes an optimal estimation of the current true status based on the current measurement. The equation of the KF is divided into two parts: the time update equation and the measurement update equation. The time update equation uses historical predictions to estimate the current state, which can be regarded as a prediction equation; the measurement update equation adjusts the estimation of the target status through the current actual measurement, which can be regarded as a correction equation. The prediction equation can be expressed as:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k, \quad (21)$$

where  $\hat{x}_{k|k-1}$  represents the target state vector predicted at time  $k$ ,  $u_k$  is the control variant, and  $B_k$  is the control matrix, which shows how the control variant  $u_k$  acts on the current state.  $\hat{x}$  is a four-dimensional vector  $[x \ y \ dx \ dy]$ , where  $x$  and  $y$  represent the coordinates of the center position of the target, and  $dx$ ,  $dy$  represent the speed of the target.  $F_k$  is the state transition matrix, which is expressed as:

$$F_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (22)$$

We set  $P_{k|k-1}$  to the covariance matrix of the posterior estimation error at time  $k$ ; the transfer process of the noise covariance matrix can be expressed as:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \quad (23)$$

where the covariance matrix  $Q_k$  represents the error caused by the prediction model at time  $k$ . In the updating equation, the KF uses the current measurement to correct the estimated results given by the prediction equation. The observed value  $Z_t$  can be expressed as:

$$Z_k = H_k \hat{x}_{k|k-1} + V_k. \quad (24)$$

The role of  $H_k$  is to transform the state space to the measurement space, and  $V_k$  is the observation noise at time  $k$ . We use  $R_k$  to represent the error covariance matrix of the observational measurement  $Z_k$ . The role of Kalman coefficient  $K_k$  is to weigh the magnitude of the predictive error covariance matrix  $P_{k|k-1}$  and the observational measurement error covariance matrix  $R_k$ , determine which of the prediction model and the observation model is more reliable, and switch the system from the observation domain to the state domain. The Kalman coefficient  $K_k$  can be expressed as:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}. \quad (25)$$

Based on what has been mentioned above, we update our KF using the following equation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}), \quad (26)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (27)$$

When the local-global tracker is tracking normally, we input the results of the trackers as observations into the KF for parameter update. When target occlusion or tracking failure occurs, we completely use KF to predict the movement of the target. We use the Peak to Sidelobe Ratio (PSR) value of the global filter response map and the maximum value of the local filter response map to judge the reliability of the tracker. The calculation of PSR is as follows:

$$PSR_l^t = \frac{\max(R_l^t) - \mu_l^t}{\sigma_l^t}, \quad (28)$$

where  $R_l^t$  is the response graph of the global filter on the  $l$ -th layer at time  $t$ ,  $\max(R^t)$  represents the maximum value of the global filter response map, and  $\mu^t$  and  $\sigma^t$  are the mean and standard deviation of the response map, respectively. When the target starts to enter the occlusion situation, the PSR value will decrease; when the target is completely occluded, the filter will learn the appearance characteristics of the occluded, and the PSR value will increase instead. In addition, different convolutional layers have different sensitivities to occlusion or rotation. The lower level convolutional features contain more spatial detail information and are more sensitive to changes in the appearance of the target. Therefore, we propose a time-space weighted PSR (TSPSR), which is calculated as follows:

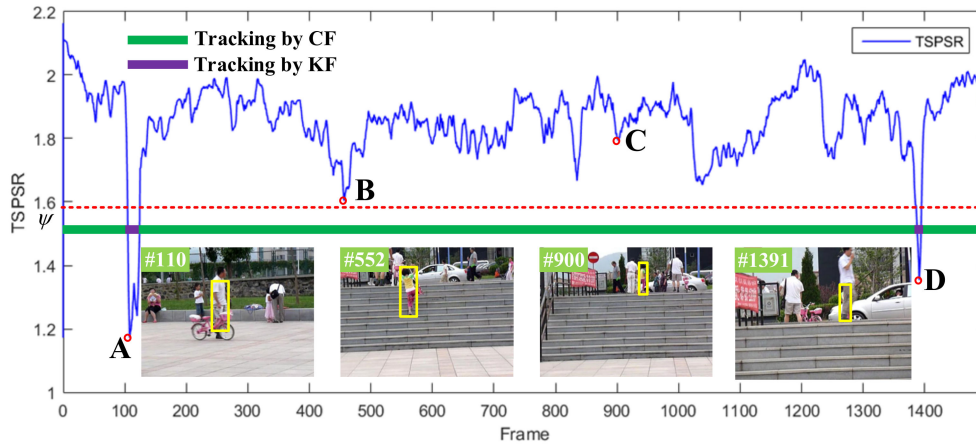
$$TSPSR^t = \sum_{t=1}^T \omega_t \frac{1}{L} \sum_{l=1}^L \beta_l PSR_l^t, \quad (29)$$

where  $\beta_l$  is the spatial weighting coefficient for the current  $l$ -th layer, and  $\omega_t$  is the time weighting coefficient for  $t$  frames before the start of the current frame. In our algorithm, the switching mechanism for the correlation filter-based tracker (CFT) and Kalman filter based tracker (KFT) is as follows:

$$\begin{cases} TSPSR^t < \psi, \forall ir_{\max}^i < r_{tre} & \text{tracking by KFT} \\ \text{otherwise} & \text{tracking by CFT} \end{cases} \quad (30)$$

where  $\psi$  is the TSPSR threshold we set. TSPSR can effectively reflect the reliability of the current correlation filter tracking results. Figure 4 intuitively reflects the distribution of TSPSR on the Girl2 sequence. In this figure, we use green to represent the stage of tracking by CF, and purple to represent the stage of tracking by KF. Points B and C indicate that the girls in frame #552 and frame #900 are not occluded, and the TSPSR value is higher than

the threshold; therefore, we use CF for tracking. However, the two points A and D indicate that the girl is occluded by a pedestrian in frames #110 and frame #1391, and the TSPSR value is lower than the threshold; therefore, we use KF for tracking.



**Figure 4.** Distributions and analysis of time-space peak to sidelobe ratio (TSPSR) value on the Girl2 sequence. The green line represents the stage of tracking by CF, and the purple line represents the stage of tracking by KF. The four points A to D correspond to different frames, and the tracking model is selected according to the value of TSPSR.

#### 4. Performance Analysis

In this section, we perform extensive experiments for the proposed algorithm using several benchmarks. First, we describe the detailed implementation and parameter settings of our tracker. Second, we analyze the effectiveness of each contribution in our proposed algorithm. Finally, we present the tracking performance of our proposed tracker compared with some state-of-the-art trackers. We perform qualitative-, quantitative-, and attributes-based evaluations on the OTB-2013 and OTB-2015 datasets to evaluate the performance of our algorithm.

##### 4.1. Experimental Setup

###### 4.1.1. Implementation Details

Our proposed tracker was implemented in MATLAB 2016b on a computer with an Intel (R) Xeon (R) CPU e5-2640 2.40 GHz CPU and a NVIDIA GeForce GTX1080Ti GPU (11 g memory) (OMNISKY, Beijing, China). The CUDA version is 10.1. We use MatConvNet which is a common deep learning toolbox in MATLAB to implement the extraction of convolutional features of VGGNet-19. More specifically, parameters of the tracker are set as follows: the kernel width of the local filter  $\sigma$  is set to 0.1, the global regularization parameter  $\lambda_g$  is set to 0.0001, and the local learning rate  $\eta_l$  is set to 0.18. We use the features of conv3-4, conv4-4, and conv5-4 of VGGNet-19 with their weights set to 0.25, 0.5, and 1, respectively. The local tracker reliability threshold  $r_{tre}$  is set to 0.55. In the TSPSR, the first five frames are taken, the spatial weight vector  $\beta_l$  is set to [0.25 0.5 0.1], the time weights vector of the previous five frames  $\omega_t$  are set to [0.4 0.2 0.2 0.1 0.1], and the confidence circle radius weight is set to 0.9.

###### 4.1.2. Datasets

In order to fully verify the effectiveness of our proposed algorithm, we perform comparative experiments on the OTB-2013 and OTB-2015 datasets, which contain 50 image sequences and 100 image sequences, respectively. These two datasets contain 11 common challenge attributes, including illumination variation (IV), occlusion (OCC), scale change (SV), fast motion (FM), deformation (DEF), motion blur (MB), out-of-view (OV), background clutter (BC), low resolution (LR), in-plane rotation (IPR), and out-of-plane rotation (OPR).

#### 4.1.3. Evaluation Indicators

The One-Pass Evaluation (OPE) proposed in OTB-2013 [25] is used to objectively evaluate the performance of the tracker, mainly using two indicators: precision and success rate. Precision is defined as the percentage of frames whose average Euclidean distance between the center position of the tracked target and ground-truth is less than the given threshold. The success rate represents the percentage of successful frames whose overlap rate between the tracked bounding boxes and ground-truth is greater than the given threshold. The compared trackers are ranked by the area under the curve (AUC).

#### 4.2. Effectiveness Analysis

##### 4.2.1. Analysis of Local–Global Multiple CF Learning Model

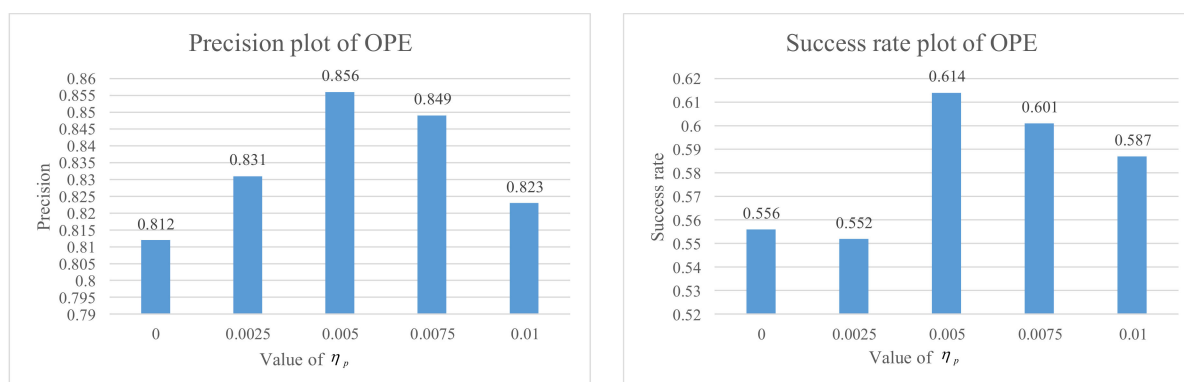
We design experiments to verify the effectiveness of our LGCF learning model, which are shown in Table 2. Global (baseline) represents that we use the global model as a baseline. Compared with a single global correlation filter model, it can be found that the local–global multiple correlative filter collaborative model achieves a 1.34% rise in accuracy and a 1.89% rise in success rate on OTB-2013, a 1.30% improvement in accuracy, and a 1.82% improvement in success rate on OTB-2015.

**Table 2.** Effectiveness study of proposed local–global cooperation mechanism and KFR.

Types of Models	OTB-2013 Precision	OTB-2013 Success Rate	OTB-2015 Precision	OTB-2015 Success Rate
Global (Baseline)	0.891	0.635	0.845	0.603
Global + Local	0.903	0.647	0.856	0.614
Global + Local + KFR	0.906	0.649	0.862	0.616

##### 4.2.2. Analysis of Collaborative Update Strategy

As is shown in Figure 5, experimental results show the effectiveness of a local–global collaborative updating strategy on OTB-2013 and OTB-2015. We conduct experiments on the different values of the updating parameter  $\eta_p$  in the partial occlusion of the global tracker.  $\eta_p$  set to 0 represents whether the target is partially or fully occluded and the global tracker will stop updating.  $\eta_p$  set to 0.01 represents only when the target is fully occluded and then the global tracker will stop updating. When the  $\eta_p$  is set to 0.005, the best performance is obtained.



**Figure 5.** The experimental results for different reliability ratio threshold  $\eta_p$  on OTB-2015.

#### 4.3. Overall Performance

We compare the proposed trackers with the other 12 state-of-the-art trackers, and we divided these trackers into the following categories: (1) DCF-based trackers: SRDCF [6], SAMF\_AT [49], MUSTer [50], DSST [33], LCT [22], Staple [51]. (2) CNN-based trackers:

HCF [17], SiamFC [35], HDT [15], CFNet [52], DeepSRDCF [13]. (3) Set-based tracker: MEEM [53].

#### 4.3.1. Quantitative Evaluation

We compare the overall accuracy and success rate of the proposed tracker with the other 12 trackers on OTB-2013 and OTB-2015. Figures 6 and 7 show the evaluation results on these two datasets. It can be seen that compared with the other latest methods, our proposed tracker shows excellent performance in both success rate and accuracy, because the OTB-2015 dataset contains the OTB-2013 dataset but includes some more challenging sequences.

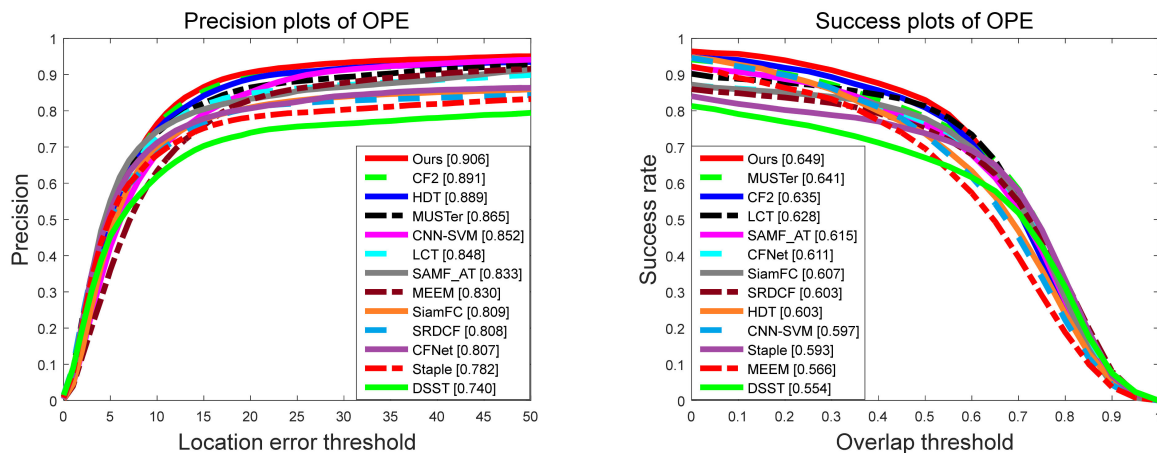


Figure 6. Precision and success rate of our proposed tracker and compared other 12 trackers on OTB-2013.

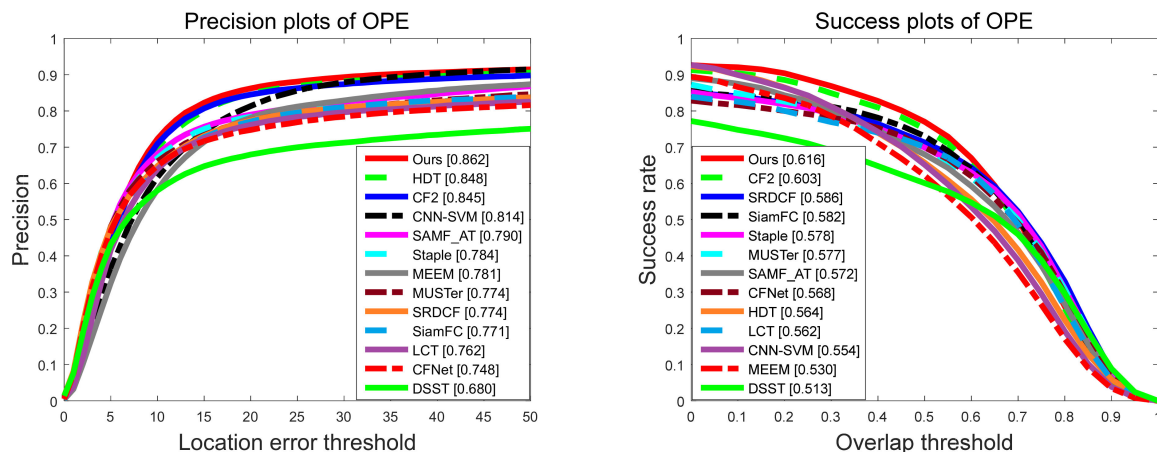
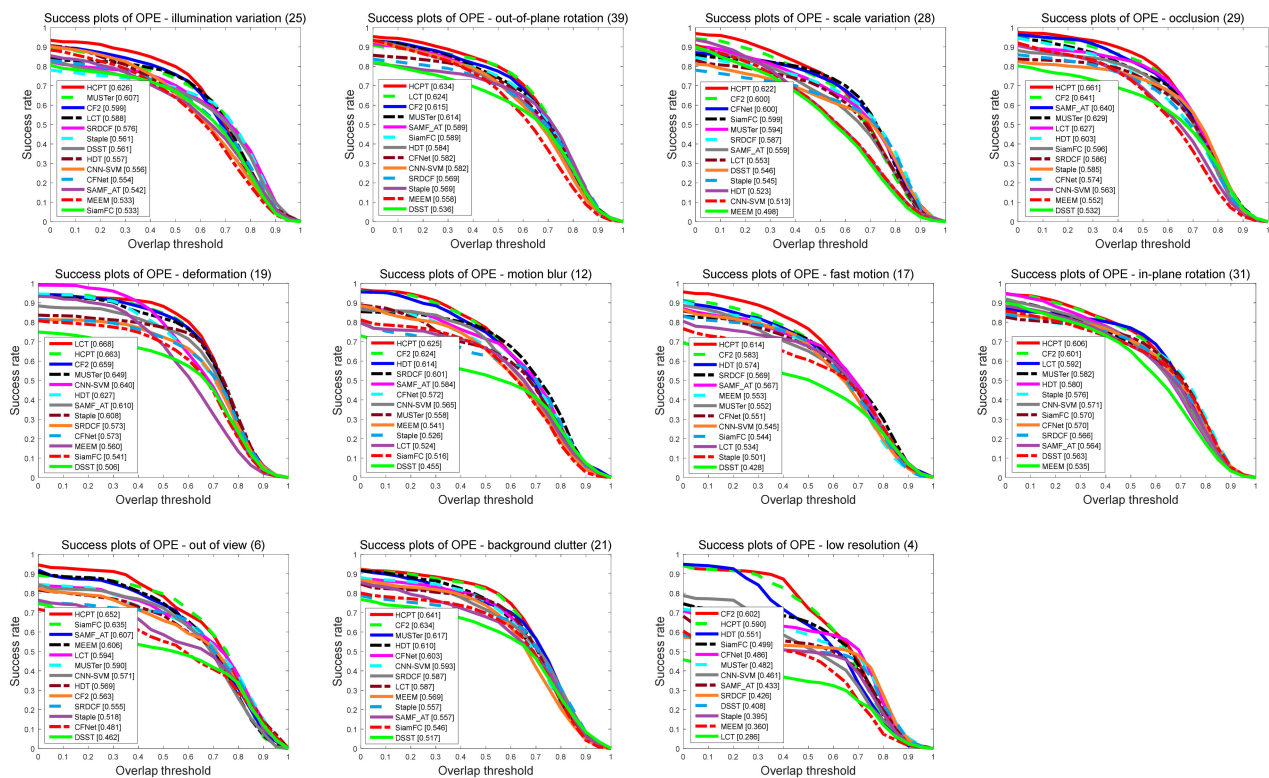


Figure 7. Precision and success rate of our proposed tracker and compared other 12 trackers on OTB-2015.

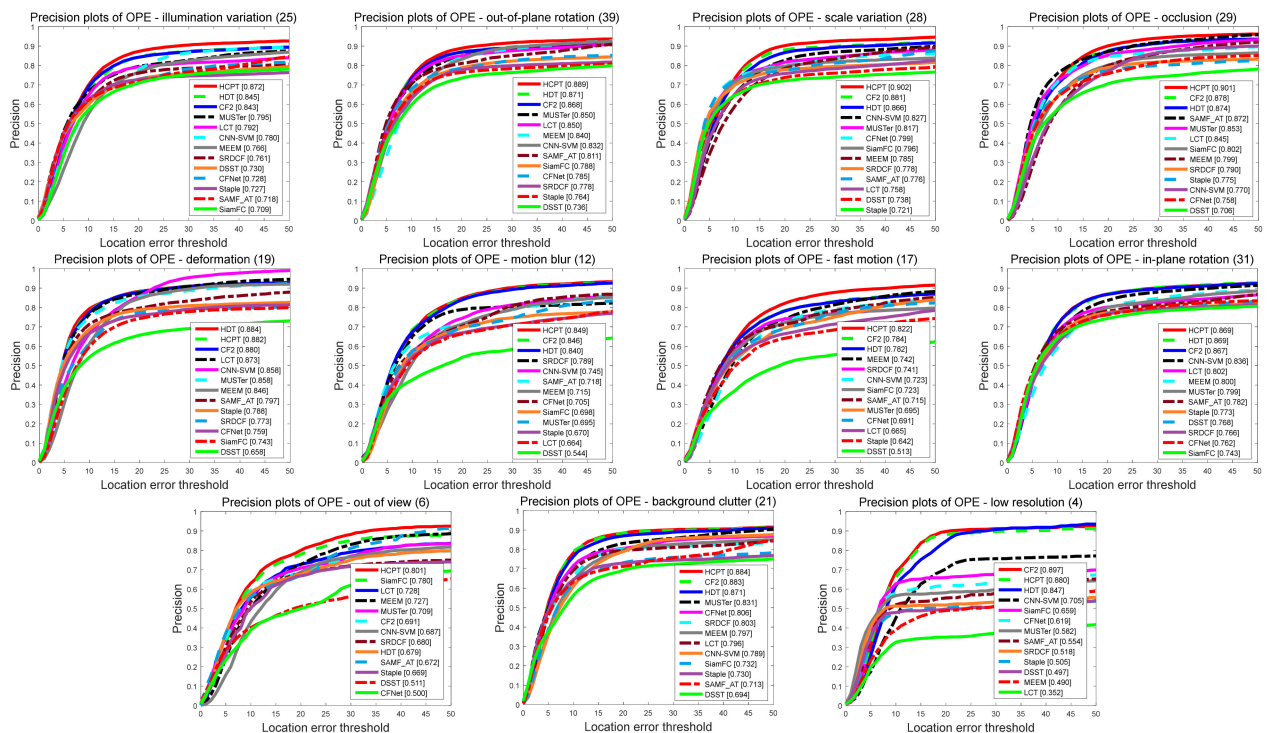
#### 4.3.2. Attribute-Based Evaluation

We use 11 challenging attributes on OTB-2013 and OTB-2015 to comprehensively evaluate the accuracy and robustness of our proposed tracker in various challenging scenarios. Figures 8 and 9 show the evaluation results of our tracker and the other 12 state-of-the-art trackers based on different attributes on OTB-2013, respectively. Table 3 shows the results of the attribute-based accuracy rate on OTB-2015. It can be seen from the figure that our proposed algorithm performs well in most of the 11 attributes, especially in the 2 attributes OV and OCC. Owing to our local tracking module and KFR module, it can effectively deal with challenging situations such as out-of-view and occlusion. In addition, the local-global collaborative updating strategy enables our filter to correctly learn the appearance of the tracking target, and avoids learning the appearance of foreground occlusions or backgrounds.





**Figure 8.** Success rate plots of 11 tracking challenging on OTB-2013. The plots illustrate the experimental results using 11 attributes. The legend shows the success rate and we show the results of 12 trackers.



**Figure 9.** Precision plots of 11 tracking challenging on OTB-2013. The plots illustrate the experimental results using 11 attributes. The legend shows the success rate and we show the results of 12 trackers.

**Table 3.** Performance evaluations of different attributes on OTB-2015. We use red and blue fonts to mark the top two scores.

Trackers	IV	OR	SV	OCC	DEF	MB	FM	IR	OV	BC	LR
CNN-SVM	0.789	0.806	0.791	0.73	0.793	0.767	0.742	0.798	0.659	0.776	0.79
SRDCF	0.786	0.724	0.75	0.703	0.699	0.782	0.762	0.739	0.572	0.775	0.631
Staple	0.776	0.738	0.739	0.728	0.751	0.719	0.709	0.772	0.644	0.749	0.591
MUSTer	0.776	0.753	0.72	0.734	0.689	0.699	0.684	0.756	0.603	0.784	0.677
LCT	0.739	0.748	0.687	0.682	0.689	0.673	0.667	0.768	0.616	0.734	0.49
MEEM	0.733	0.795	0.74	0.741	0.754	0.722	0.728	0.79	0.690	0.746	0.605
SiamFC	0.728	0.75	0.739	0.722	0.69	0.724	0.741	0.752	0.65	0.69	0.815
SAMF_AT	0.723	0.764	0.75	0.748	0.687	0.765	0.718	0.778	0.652	0.713	0.716
DSST	0.714	0.641	0.655	0.597	0.542	0.595	0.562	0.689	0.483	0.704	0.55
CFNet	0.686	0.728	0.715	0.674	0.643	0.687	0.695	0.759	0.572	0.724	0.787
HDT	0.815	0.807	0.812	0.774	0.821	0.794	0.802	0.834	0.687	0.844	0.766
HCF	0.835	0.817	0.807	0.776	0.791	0.804	0.792	0.849	0.689	0.852	0.822
Ours	0.87	0.85	0.835	0.817	0.806	0.846	0.796	0.856	0.784	0.851	0.867

\* For the scores in Table 3, 11 columns represent the performance of different algorithms under 11 challenge attributes, and 13 rows represent the performance of 13 algorithms under different challenge attributes. The best and the second-best precision in each column are highlighted in red and blue respectively.

#### 4.3.3. Qualitative Evaluation

Figure 10 shows the tracking results of our proposed tracker and five other trackers (including Staple [51], HDT [15], CFNet [52], SiamFC [35], and HCF [17]) on ten challenging video sequences. These frame sequences are from the public benchmarks OTB-2013 [25] and OTB-2015 [26]. In general, our tracker can track targets more accurately. In the “Box” sequence, the target is almost completely occluded at the 47th frame, and our tracker can still track it accurately. At the 508th frame, only our tracker can track the object correctly. In the “Girl2” sequence, when the little girl is completely occluded by a pedestrian and then appears in view, only our tracker can accurately recognize it, which contributes to our KFR mechanism. For “Lemming” sequences with OCC, IPR, and IR challenges, occlusion at the 372nd frame forces most trackers to lose their targets. In the 408th frame, only our tracker and SiamFC could accurately track the target. For “Soccer”, “Skating1”, and “Ironman” sequences with IV and BC, our tracker copes with illumination variation and background clutter very well. For “Dragonbaby” sequences with OPRs and OV challenges, SiamFC and CFNet have lost targets, but our tracker is still able to accurately capture targets. At the same time, as shown in the “Trellis 1” sequence with the SV challenge, our tracker can also handle scale changes very well. For “Clifbar” and “MotorRolling” sequences with BC, IV, and FB challenges, the other five methods perform poorly, but our tracker can handle this complex situation.





**Figure 10.** The comparisons of bounding box for different algorithms on 7 challenging image sequences of OTB-2013 and OTB-2015 (from top to down are Box, Girl2, Lemming, ClifBar, Ironman, DragonBaby, and Soccer, respectively).

## 5. Conclusions

This paper proposes a local–global multiple correlation filters-based object tracking algorithm. First of all, global filters and local filters transferring information to each other can effectively cope with the challenges of occlusion and rotation with the comprehensive use of convolutional features and hand-crafted features. Based on our learning model, we propose a local–global collaborative updating strategy, which can prevent the tracker from learning the appearance of foreground occlusions or background clutter. In addition, we propose the time-space peak to sidelobe ratio (TSPSR) index, which can effectively reflect the reliability of the current KF tracker. When the current KF fails, the KFR model can be used to accurately recapture the target and predict the position of the original target. The experimental results on the OBT-2013 and OTB-2015 show that our tracker performs favorably against the other latest 12 trackers over accuracy and robustness.

**Author Contributions:** Conceptualization, J.Z. and Y.L.; methodology, J.Z., Y.L., J.W.; software, Y.L.; validation, H.L.; data curation, J.Z., Y.L.; writing—original draft preparation, J.Z., Y.L. and J.W.; writing—review and editing, J.Z. and Y.L.; visualization, Y.L.; supervision, J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 61972056, in part by the Basic Research Fund of Zhongye Changtian International Engineering Co., Ltd. under Grant 2020JCYJ07, in part by the “Double First-class” International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology under Grant 2019IC34, in part by the Postgraduate Training Innovation Base Construction Project of Hunan Province under Grant 2019-248-51, and in part by the Postgraduate Scientific Research Innovation Fund of Hunan Province under Grant CX20190695.

**Institutional Review Board Statement:** Ethical review and approval were waived for the above-mentioned study, because all the images used in the manuscript and its experiments can be downloaded from [http://cvlab.hanyang.ac.kr/tracker\\_benchmark/](http://cvlab.hanyang.ac.kr/tracker_benchmark/) website for free.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: [http://cvlab.hanyang.ac.kr/tracker\\_benchmark/datasets.html](http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

- Adam, A.; Rivlin, E.; Shimshoni, I. Robust Fragments-Based Tracking Using the Integral Histogram. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; IEEE Xplore: New York, NY, USA, 2006; Volume 1, pp. 798–805.
- Bao, C.; Wu, Y.; Ling, H.; Ji, H. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–17 June 2012; IEEE Xplore: New York, NY, USA, 2012; pp. 1830–1837.
- Babenko, B.; Yang, M.H.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1619–1632. [[CrossRef](#)] [[PubMed](#)]
- Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Li, K. Dual model learning combined with multiple feature selection for accurate visual tracking. *IEEE Access* **2019**, *7*, 43956–43969. [[CrossRef](#)]
- Bolme, D. Visual Object Tracking Using Adaptive Correlation Filters. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
- Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 4310–4318.
- Chen, Y.; Wang, J.; Xia, R.; Zhang, Q.; Cao, Z.; Yang, K. The visual object tracking algorithm research based on adaptive combination kernel. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 4855–4867. [[CrossRef](#)]
- 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](#)]
- Comaniciu, D.; Ramesh, V.; Meer, P. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. In Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June 2000; IEEE Xplore: New York, NY, USA, 2000; pp. 142–149.
- Tuytelaars, T.; Mikolajczyk, K. Local Invariant Feature Detectors: A Survey. In *Foundations and Trends in Computer Graphics and Vision*; now Publisher: Hanover, MA, USA, 2008; Volume 3, pp. 177–280.
- Viola, P.; Jones, M.J. Robust Real-Time Face Detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [[CrossRef](#)]
- Wang, S.; Lu, H.; Yang, F.; Yang, M.H. Superpixel Tracking. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1323–1330.
- Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 13–16 December 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 58–66.
- Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Sangaiah, A.K. Spatial and semantic convolutional features for robust visual object tracking. *Multimed. Tools Appl.* **2020**, *79*, 15095–15115. [[CrossRef](#)]



15. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, J.; Lim, J.; Yang, M.H. Hedged Deep Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE Xplore: New York, NY, USA, 2016; pp. 4303–4311.
16. Zhang, J.; Wu, Y.; Feng, W.; Wang, J. Spatially attentive visual tracking using multi-model adaptive response fusion. *IEEE Access* **2019**, *7*, 83873–83887. [[CrossRef](#)]
17. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 13–16 December 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 3074–3082.
18. He, Z.; Fan, Y.; Zhuang, Y.; Dong, Y.; Bai, H. Correlation Filters with Weighted Convolution Responses. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE Xplore: New York, NY, USA, 2017; pp. 1992–2000.
19. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.; Yang, M.H. Crest: Convolutional Residual Learning for Visual Tracking. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE Xplore: New York, NY, USA, 2017; pp. 2555–2564.
20. Zhang, J.; Sun, J.; Wang, J.; Yue, X.G. Visual object tracking based on residual network and cascaded correlation filters. *J. Ambient Intell. Humaniz. Comput.* **2020**. [[CrossRef](#)]
21. Akin, O.; Erdem, E.; Erdem, I.A.; Mikolajczyk, K. Deformable part-based tracking by coupled global and local correlation filters. *J. Vis. Commun. Image Represent.* **2016**, *38*, 763–774. [[CrossRef](#)]
22. Ma, C.; Yang, X.; Zhang, C.; Yang, M.H. Long-Term Correlation Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 5388–5396.
23. Jing, X.Y.; Zhu, X.; Wu, F.; You, X.; Liu, Q.; Yue, D.; Hu, R. Super-Resolution Person Re-Identification with Semi-Coupled Low-Rank Discriminant Dictionary Learning. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 695–704.
24. Lai, Z.; Wong, W.K.; Xu, Y.; Yang, J.; Zhang, D. Approximate orthogonal sparse embedding for dimensionality reduction. *IEEE Trans. Neural Networks Learn. Syst.* **2015**, *27*, 723–735. [[CrossRef](#)] [[PubMed](#)]
25. 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, Portland, OR, USA, 23–28 June 2013; IEEE Xplore: New York, NY, USA, 2013; pp. 2411–2418.
26. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [[CrossRef](#)] [[PubMed](#)]
27. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the 12th European Conference on Computer Vision, Firenze, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.
28. Zhang, J.; Wang, W.; Lu, C.; Wang, J.; Sangaiah, A.K. Lightweight deep network for traffic sign classification. *Ann. Telecommun.* **2020**, *75*, 369–379. [[CrossRef](#)]
29. Zhang, J.; Lu, C.; Li, X.; Kim, H.J.; Wang, J. A full convolutional network based on DenseNet for remote sensing scene classification. *Math. Biosci. Eng.* **2019**, *16*, 3345–3367. [[CrossRef](#)] [[PubMed](#)]
30. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; IEEE Xplore: New York, NY, USA, 2017; Volume 7, pp. 6638–6646.
31. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 472–488.
32. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the 8th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 254–265.
33. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Accurate Scale Estimation for Robust Visual Tracking. In Proceedings of the 25th British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; BMVA Press: Durham, UK, 2014; Volume 7, pp. 83873–83887.
34. Li, Y.; Zhu, J.; Hoi, S.C. Reliable Patch Trackers: Reliable Patch Trackers: Robust Visual Tracking by Exploiting Reliable Patches. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 353–361.
35. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 850–865.
36. Guo, Q.; Feng, W.; Zhou, C.; Huang, R. Learning Dynamic Siamese Network for Visual Object Tracking. In Proceedings of the 2017 IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE Xplore: New York, NY, USA, 2017; pp. 1763–1771.



37. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; IEEE Xplore: New York, NY, USA, 2018; pp. 8971–8980.
38. Wang, Q.; Zhang, L.; Bertinetto, L. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; IEEE Xplore: New York, NY, USA, 2019; pp. 1328–1338.
39. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K. A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE Xplore: New York, NY, USA, 2009; pp. 248–255.
40. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:abs/1409.1556.
41. Ba, S.; He, Z.; Xu, T.B.; Zhu, Z.; Dong, Y.; Bai, H. Multi-Hierarchical Independent Correlation Filters for Visual Tracking. *arXiv* **2018**, arXiv:abs/1811.10302v1.
42. Sun, Y.; Xie, J.; Guo, J.; Wang, H.; Zhao, Y. A Modified Marginalized Kalman Filter for Maneuvering Target Tracking. In Proceedings of the 2nd International Conference on Information Technology and Electronic Commerce, Zurich, Switzerland, 14–15 June 2014; pp. 107–111.
43. Divya, V.X.; Kumari, B.L. Performance Analysis of Extended Kalman Filter for Single and Multi Target Tracking. In Proceedings of the 13th International Conference on Electromagnetic Interference and Compatibility, Visakhapatnam, India, 22–23 July 2015; pp. 53–57.
44. Bukey, C.M.; Kulkarni, S.V.; Chavan, R.A. Multi-Object Tracking Using Kalman Filter and Particle Filter. In Proceedings of the 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, Chennai, India, 21–22 September 2017; IEEE Xplore: New York, NY, USA, 2017; pp. 1688–1692.
45. Patel, H.A.; Thakore, D.G. Moving object tracking using Kalman filter. *Int. J. Comput. Sci. Mob. Comput.* **2013**, *2*, 326–332.
46. Naresh Boddeti, V.; Kanade, T.; Vijaya Kumar, B. Correlation Filters for Object Alignment. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; IEEE Xplore: New York, NY, USA, 2013; pp. 2291–2298.
47. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [[CrossRef](#)] [[PubMed](#)]
48. Van de Weijer, J.; Schmid, C.; Verbeek, J.; Larlus, D. Learning color names for real-world applications. *IEEE Trans. Image Process.* **2009**, *18*, 1512–1523. [[CrossRef](#)] [[PubMed](#)]
49. Bibi, A.; Mueller, M.; Ghanem, B. Target Response Adaptation for Correlation Filter Tracking. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 419–433.
50. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. Multistore Tracker (Muster): A Cognitive Psychology Inspired Approach to Object Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 749–785.
51. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE Xplore: New York, NY, USA, 2016; pp. 1401–1409.
52. 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 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; IEEE Xplore: New York, NY, USA, 2017; Volume 7, pp. 2805–2813.
53. Lee, D.Y.; Sim, Y.J.; Kim, C.S. Multihypothesis Trajectory Analysis for Robust Visual Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; IEEE Xplore: New York, NY, USA, 2015; pp. 5088–5096.