algorithms

MDPI

Article

A Robust Visual Tracking Algorithm Based on Spatial-Temporal Context Hierarchical Response Fusion

Wancheng Zhang [1], Yanmin Luo [2,3], Zhi Chen [1], Yongzhao Du [1], Daxin Zhu [4] and Peizhong Liu [1,*]

[1] College of Engineering, Huaqiao University, Quanzhou 362021, China; wancheng_z@126.com (W.Z.); marico2018@163.com (Z.C.); yongzhaodu@126.com (Y.D.)
[2] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; lym@hqu.edu.cn
[3] The Key Laboratory for Computer Vision and Pattern Recognition, Huaqiao University, Xiamen 361021, China
[4] School of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362021, China; huiyi@qztc.edu.cn
* Correspondence: pzliu@hqu.edu.cn; Tel.: +86-595-2333-9012

Received: 17 October 2018; Accepted: 19 December 2018; Published: 26 December 2018

Abstract: Discriminative correlation filters (DCFs) have been shown to perform superiorly in visual object tracking. However, visual tracking is still challenging when the target objects undergo complex scenarios such as occlusion, deformation, scale changes and illumination changes. In this paper, we utilize the hierarchical features of convolutional neural networks (CNNs) and learn a spatial-temporal context correlation filter on convolutional layers. Then, the translation is estimated by fusing the response score of the filters on the three convolutional layers. In terms of scale estimation, we learn a discriminative correlation filter to estimate scale from the best confidence results. Furthermore, we proposed a re-detection activation discrimination method to improve the robustness of visual tracking in the case of tracking failure and an adaptive model update method to reduce tracking drift caused by noisy updates. We evaluate the proposed tracker with DCFs and deep features on OTB benchmark datasets. The tracking results demonstrated that the proposed algorithm is superior to several state-of-the-art DCF methods in terms of accuracy and robustness.

Keywords: visual tracking; discriminative correlation filters; hierarchical convolutional features; re-detection; model update

1. Introduction

Visual object tracking is a basic task in computer vision, with a wide range of applications such as autonomous driving, robotics, video surveillance, human-machine interaction and so forth [1,2]. Although the initial frame of the target is given, how to use an effective method to judge the position of the target in the subsequent frame is a difficult problem. These methods should be able to overcome various challenges well, including background clutter, illumination changes, scale variation, motion blur, and partial occlusions.

In recent years, Discriminative Correlation Filter (DCF) based tracking methods have shown prominent results on object tracking benchmarks [3–6]. The discriminative methods view the tracking task as a binary classification problem. During the tracking process, a binary classifier is learned online to distinguish the target and its surrounding background, and the learned classifier is used to classify the image blocks in the current image frame-mark whether the pixel belongs to the target or the background. The main goal is to find the area with the highest confidence for classifier, which is the target location, and to use the tracking result as a sample to update classifier. This method is also

called tracking-by-detection approaches. Our work will follow the DCFs tracking methods based on the tracking-by-detection framework.

Moreover, deep convolutional neural networks (CNNs) have shown high performance in many tasks. Activations from the last convolutional layers have been successfully employ for image classification [7–9]. Features from these deep convolutional layers are effective in saving spatial and structural information of the object. Ma et al. [10] proposed the use of hierarchical convolutional features in VGGNet [8] for visual tracking. The main tracking task is to extract and use the features of each convolution layer. On the one hand, shallow features can accurately locate the target, but its disadvantage is that it does not capture semantic information very well. On the contrary, the advantages of deep features can capture semantic information very well. The disadvantage is that it cannot describe exhaustive spatial details to locate the target. The semantic information has a great effect on the object after the appearance changes. After the appearance changes, the semantic information has a great effect on the tracking. Therefore, the deep features of CNNs play an increasingly important role in visual object tracking. In this paper, we learn the above mentioned methods to extract hierarchical convolutional features as feature representation.

We learn a spatial-temporal context correlation filter on convolutional layers and employ these correlation response scores for fusion to estimate the location of the target. Zhang et al. [11] proposed a spatial-temporal relationship between the target and its surrounding context regions, indicating that the context information of the target and its surrounding background regions can effectively improve tracking results. Ma et al. [12] showed that the correlation between spatial-temporal contexts can improve tracking accuracy and robustness. It is necessary to establish the spatial-temporal relationship between the target and its surrounding environment. Therefore, we employ a context-aware framework [13] based on discriminative correlation filter as our spatial-temporal context model. On this basis, we obtain a powerful filter that produces a high response value for the target image block and a near-zero response value for the context region. In order to estimate scale changes adaptively, we utilize the HOG feature as feature representation to train a discriminative correlation filter and estimate the desired object scale from the best score frame. HOG [14] feature is a feature descriptor that used for object detection in computer vision and image processing. It has certain translation invariance, rotation invariance, and illumination invariance, which can better adapt to target deformation, scale changes, and occlusion.

It is very important to design an effective re-detection method to improve the robustness of visual tracking in the case of tracking failure. In this work, we employ the EdgeBox [15] to achieve object online re-detection and use the predefined threshold as the activation condition for re-detection. However, it did not perform very well for all the video sequences. To this end, we proposed a self-adaptive activation method to stimulate the re-detection component. We compared the size of the response map peak and its corresponding peak-to-side lobe ratio (*PSR*) [5] score generated by DCFs. By this method, the detector can be well awakened when the condition is satisfying (Section 3.3.1). For the model updating, most existed tracking algorithms often update the tracking model at a fixed interval or frame-by-frame [6,12,16–18]. These approaches have some obvious disadvantages. If objects go through complex appearance changes for instance occlusion and disappear in the current frame, these situations bring will fault background information. The wrong information is delivered to subsequent frames and decrease the performance of tracking after accumulating for a long time. The end result is tracking drift. Hence, we propose an effective model updating method similar to re-detection method. We compare the size of the response map peak and its corresponding *PSR* score generated by DCFs. By this method, the tracking models can be updated in time to improve tracking robustness and avoid tracking drift effectively due to noisy problems (Section 3.4).

The main contributions of this work are as follows:

(1) The hierarchical features of CNNs are used as feature representation to handle large appearance variations, and we learn a spatial-temporal context correlation filter on each CNN layer as a discriminative classifier. We use multi-level correlation response maps for fusion to infer the

target location. For scale estimation, we train DCF based on scale pyramid representation and estimate the desired object scale from the best score frame.

(2) We employ the EdgeBox to redetect when tracking failure occurred and proposed a novel re-detection activation method. For model updating, we propose a novel model update method to solve the model noisy problems.

(3) We extensively validate our method on benchmark datasets with large-scale sequences and extensive experimental results demonstrated that the proposed tracking algorithm is superior to the state-of-the-art methods in terms of accuracy and robustness.

2. Related Works

In Discriminative Correlation Filter based trackers, the filter is trained to predict the optimal response map by minimizing a least-squares loss for all circular shifts of a training sample. Since the complicated convolution operations can be converted into simple element-wise multiplication operations, DCF shows the advantage of high computational efficiency. Firstly, Bolme et al. [5] proposed a tracker using minimum output sum of squared error (MOSSE) filter, which uses a grayscale template. Henriques et al. [19] replaced the grayscale templates by HOG [14] features and built on multiple channel features which further improved the tracking accuracy and robustness. Danelljan et al. [16] learned separate filters for translation and scaling. The role of the two filters is to locate the target and estimate desired scale of the target object, respectively. Zhang et al. [20] incorporated context information into filter learning. Luca et al. [17] proposed the STAPLE tracker which combines DCF and color histogram based model. Danelljan et al. [21] introduced a spatial regularization component in the learning to punish correlation filter coefficients depending on their spatial location, which enhanced the robustness of tracking effectively. Wang et al. [22] proposed a large margin visual tracking method with circulant feature maps, which employed a multi-modal detection technique to avoid tracking drift. C-COT [23] and ECO [24] adopted an implicit interpolation model to solve the learning problems in the continuous spatial domain, which enhanced the tracking accuracy. Alam et al. [25] introduced a new metric called the peak-to-clutter mean (PCM) and it provided sharp and high correlation peaks corresponding to targets. This method improves the efficiency of detection. Paheding Sidike et al. [26] introduced class-associative spectral fringe-adjusted joint transform correlation (CSFJTC) based on joint transform correlation (JTC) and employed class-associative filtering, modified Fourier plane image subtraction, and fringe-adjusted JTC techniques to execute the object detection task. The performance of the detection was outstanding. To reduce the training time significantly for online training of the object, Evan Krieger et al. [27] proposed Progressively Expanded Neural Network (PENNet) tracker methodology and employed a modified variant of the extreme learning machine. To overcome these challenges, such as object structural information distortions and background variations, Krieger et al. [28] proposed a Directional Ringlet Intensity Feature Transform (DRIFT) method, which utilized Kirsch kernel filtering for edge features and a ringlet feature mapping for rotational invariance. This method obtained accurate object boundaries and improvements for lowering computation times. Zhang et al. [29] introduced a spatial alignment module, which provides continuous feedback for transforming the target from the border to the center with a normalized aspect ratio. This method can handle undesired boundary effects. Song et al. [30] used an adversarial learning method to maintain the most robust features of the target objects and proposed a high-order cost sensitive loss to decrease the effect of easy samples.

Visual representations are important for visual tracking. Most tracking algorithms have recently employed deep features extracted from convolutional neural networks CNN as feature representation. Danelljan et al. [31] used the deep features learned from CNN for representation based DCF framework, which improved tracking performances. Fan et al. [32] learned a feature extractor with convolutional neural networks from an offline training set for visual tracking. To handle long training times and a large number of training samples, DeepTrack [33] employed a single CNN for learning effective feature representations of the target object by a purely online manner. CNT [34] proposed

a convolutional neural network model tracking framework without pre-training and used simple two-layer convolutional networks to learn robust representations for visual tracking, which improved tracking accuracy. Song et al. [35] integrated features extraction, response map generation, and model updating into the convolutional neural network for end-to-end training, which effectively improved the tracking robustness. Zhu et al. [36] proposed a joint convolutional tracking, which viewed the process of feature extraction and tracking as convolution operation. Yao et al. [37] proposed a learning representation and truncated inference model by modeling the representor as CNN and achieved competitive accuracy. At present, researchers have seen the advantages of deep networks. Some existing tracking methods [10,23,24,35,38] use fewer convolutional layers to extract target features and improve tracking robustness effectively. Therefore, we make full use of hierarchical feature of convolution layers as feature representation.

3. Robust Spatial-temporal Context Hierarchical Response Fusion

3.1. The Overall Flowchart of The Proposed Algorithm

The proposed framework for visual tracking algorithm with spatial-temporal context hierarchical response fusion is showed in Figure 1. As shown in Figure 1, we decompose the tracking task into translation estimation and scale estimation. We first extract hierarchical features of convolution layers and learn spatial-temporal context correlation filter on each layer. Then, we fuse the correlation response map of the translation filters to infer the target position. We predict the scale change using the scale filter. The re-detection module built on EdgeBox. When the *PSR* values are lower than its corresponding response map peak value R_{max}, we activate the re-detection module. When the *PSR* values were greater than its corresponding response map peak value R_{max}, we update the tracking model.

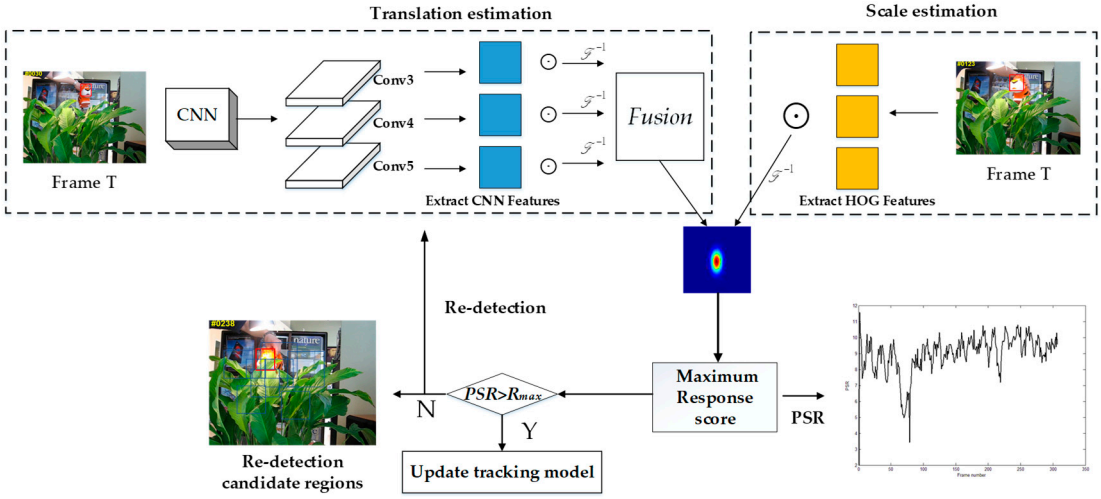


Figure 1. The overall framework of the proposed visual tracking algorithm.

3.1.1. Hierarchical Feature of Convolution Layer

The hierarchical features of deep neural networks play an important role in visual object tracking, which can enhance the robustness and accuracy of visual tracking. Existing tracking methods based on deep learning [33,34,39] find that the deep features of the convolutional layer encode the semantic information of targets, which is invariant to the appearance change. But, when the convolution layer is getting deeper, the resolution of the image is lower and it is more difficult to estimate the location of the target. In contrast, features from shallow convolutional layers capture more fine-grained spatial details and locate the target accurately but not robust to appearance changes. Our aim is to make full use of

the semantic information features of deep convolutional layers to solve large appearance changes and to utilize the features of shallow layers to locate the target precisely and prevent tracking drift.

We make use of the convolution features of a CNN (VGGNet [8]) to encode target appearance. For visual tracking, the result we want is to more accurately locate the target and know where it is located. However, the semantic information has a great effect on the object after the appearance changes and can estimate the approximate position of the tracking object. Therefore, the target can be positioned more accurately. In our work, the feature extractor we use is pre-trained VGGNet-16 [8] and we select the third, fourth and fifth layer of convolutional layers to represent target objects (Figure 1). The characteristic of conv5 layer handles serious appearance changes but does not accurately locate the target due to low resolution. On the contrary, the characteristic of conv4 and conv3 layers can capture more space details and help better locate the object. The peculiarity is used for image segmentation and detailed localization using CNNs [40]. If we can make clever use of hierarchical feature of convolution layer, it will be very helpful for our experiments in the future.

3.1.2. Spatial-Temporal Context Correlation Filter

Traditional discriminative correlation filters use cosine windows to handle the boundary effects of tracking targets due to the circulant assumption, which causes limited contextual information for DCFs based trackers. It is easily result in tacking drift when target objects experience complex scene changes such as occlusion, background clutter and fast motion. In order to learn a powerful filter that produces a high response value for the target image block and a near zero response value for the context region, the CACF framework [13] combined the background information around the target into the learned filter. In our work, we utilize the hierarchical features of convolutional layer as feature representation and employ CACF framework as our spatial-temporal context model for visual tracking, reference [13].

In this framework, we aim to get an ideal correlation filter w. For all training samples U_0 generated by circular shifts using a sliding window on three convolution layer, utilizing the nature of the circular matrix [6] will better dealt better with the ridge regression trouble.

$$\min_{w} \|U_0 w - y\|_2^2 + \lambda_1 \|w\|_2^2. \tag{1}$$

In Equation (1), the data matrix U_0 represents all circular shifts of the vectorized image patch u_0, and w denotes the learned correlation filter. The regression target y represents a vectorized image of a 2D Gaussian, λ_1 represents regularization weight parameters.

We use the learned correlation filter w to convolve with the image block in the next frame in order to predict the position of the target. The maximum response value of all training sample response vectors $y_p(z, w)$ is the estimated position of the target. Given an image block z, the output response value is derived from the following formula:

$$f(z) = \mathcal{F}^{-1}(z \odot w) = \hat{z} \odot \hat{w}. \tag{2}$$

In Equation (2), $\mathcal{F}^{-1}$ represents the inverse Fourier transformation, and $\odot$ represents the convolution operation. Then update filter model by employing following equations:

$$\hat{w}_i = \eta w_i + (1 - \eta)\hat{w}_{i-1}, \tag{3a}$$

$$\hat{x}_i = (1 - \eta)\hat{x}_{i-1} + \eta \hat{x}_i. \tag{3b}$$

The subscript i represents the sequence number of current frame, and η represents the learning rate parameter, and $\hat{x}_i$ represents the target appearance model.

3.1.3. Multi-Response Maps Fusion

The context-aware filter mainly integrates the context background information around the target into the filter to learn together and obtains a correlation filter with high discriminative performance through training. The advantage is that it can effectively utilize the context information of the surrounding area of the target and can make the target a better robustness in complex scenes such as occlusion, background clutter, and fast motion.

In order to improve tracking robustness and take full advantage of hierarchical features and each layer of filters, we use context-aware correlation filters to output response values on the third, fourth and fifth convolutional layers, respectively, recorded as $R_{context3}$, $R_{context4}$, $R_{context5}$, and then calculate the weight of each layer's response map normalized in t frame:

$$context3\_\overline{w}_t = \frac{\max(R_{context3})}{\max(R_{context3} + R_{context4} + R_{context5})}, \tag{4a}$$

$$context4\_\overline{w}_t = \frac{\max(R_{context4})}{\max(R_{context3} + R_{context4} + R_{context5})}, \tag{4b}$$

$$context5\_\overline{w}_t = \frac{\max(R_{context5})}{\max(R_{context3} + R_{context4} + R_{context5})}. \tag{4c}$$

The filter response map accounts for a larger proportion and assigns a higher weight. Update the original response weight with the weight of the t frame:

$$context3\_w_{t+1} = (1-\tau) \times context3\_w_t + \tau \times context3\_\overline{w}_t, \tag{5a}$$

$$context4\_w_{t+1} = (1-\tau) \times context4\_w_t + \tau \times context4\_\overline{w}_t, \tag{5b}$$

$$context5\_w_{t+1} = (1-\tau) \times context5\_w_t + \tau \times context5\_\overline{w}_t. \tag{5c}$$

Here, τ is the weight update parameter; and $context3\_w_t$, $context4\_w_t$ and $context5\_w_t$ denote the original response weight in t frame. At t frame, the final response is obtained by fusing each response map ($R_{context3}$, $R_{context4}$, $R_{context5}$):

$$\begin{gathered} R_t = \\ context3\_w_t \times context3\_R_t \\ +context4\_w_t \times \ context4\_R_t \\ +context5\_w_t \times context5\_R_t \end{gathered} \tag{6}$$

3.2. The Scale Discriminative Correlation Filter

When tracking the target, the target object experiences different scenes, and its appearance and scale will change with time. This situation brings great problems to the tracking, how to update the target appearance and scale change effectively in a timely manner, which is the key to improving tracking performance. We found that [16] proposed an accurate scale estimation method. This method can estimate the target scale from the best score by training a scale discriminative correlation filter. Based on this, we learn a scale discriminative correlation filter to handle the problem of target scale change and get an ideal scale correlation filter h by the following function:

$$\varepsilon = \| \sum_{l=1}^{d} h^l * f^l - g \|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2. \tag{7}$$

In Equation (7), g represents the optimal correlation output, l represents the dimension of the feature, and λ is a regular coefficient. The above solution in the frequency domain is given by:

$$H^l = \frac{\overline{G}F^l}{\sum_{k=1}^{d} \overline{F}^k F^k + \lambda} = \frac{D_t^l}{E_t}. \tag{8}$$

To better calculate the results, the numerator and denominator of H^l in Equation (8) are respectively updated as follows:

$$D_t^l = (1-\eta)D_{t-1}^l + \eta \overline{G}_t F_t^l \tag{9a}$$

$$E_{\mathrm{t}} = (1-\eta)E_{t-1} + \eta \sum_{k=1}^{d} \overline{F}_t^k F_t^k \tag{9b}$$

η represents a learning rate parameter. the response value of the scale filter that we need can be calculated by following equation:

$$\hat{y}_{\mathrm{s}} = \mathcal{F}^{-1}\left\{ \frac{\sum_{l=1}^{d} \overline{D}^l Z^l}{E + \lambda} \right\} \tag{10}$$

We estimate the target scale by getting the maximum scale response value and utilize Equations (9a) and (9b) to update the scale discriminative filter model.

3.3. Target Recovery

It is clearly known that a re-detection module is essential for realizing a robust visual tracking. To better cope with tracking failures and continue tracking in subsequent frames, we employ the EdgeBox [15] as our detector and use this method to generate candidate regions: candidate regions detection C_d is across the entire image and has large step size. In addition, we need to calculate the confidence value of candidate c and use a traditional learning rate to learn spatial-temporal context correlation filters in order to maintain a long-term memory of the target appearance. Give a candidate c, we represent the maximum response value of correlation filter by $g(c)$.

We consider the response map peak as a dynamic threshold with comparison to corresponding *PSR* score (Section 3.3.1). Only when the *PSR* score is lower than its corresponding response map peak score R_{max} and the tracking failure occurs at this time, will it proceed with target re-detection. We produce a set of candidate regions across the whole frame for recovering target objects. We choose the desired candidate as our re-detection result by minimizing the following issue:

$$\mathrm{argmin}_{\mathrm{i}}\ g(c_t^i) + \alpha D(c_t^i, c_{t-1}) \tag{11}$$

In Equation (11), the purpose of the weight factor α is to get a balance between candidate regions value and motion smoothing. D denotes the center location distance between each candidate c_t^i and the bounding box c_{t-1}.

3.3.1. Detector Activation

When the tracking failure occurs, it is very robust for the tracking result as the detector can be stimulated in time to re-detect. In this paper, we adopt the peak-to-side lobe ratio (termed *PSR*) referred by [5] to be our tracking quality evaluation. The tracking quality is evaluated according to the strength of the response map peak. The higher the *PSR* score, the more excellent the tracking quality. The *PSR* is defined as follows:

$$PSR = \frac{R_{\mathrm{max}} - \mu_{s1}}{\sigma_{s1}} \tag{12}$$

where R_{max} is the peak value of the response map R_t. The subscript $s1$ is the peak side lobe region around the peak, which is 15% of the response map area in this paper, μ_{s1} and σ_{s1} are the mean value and standard deviation of the side lobe area.

Figure 2 illustrates the distribution of the response map generated by DCFs and its corresponding *PSR* score. It is clear that the tracking perform well when the *PSR* scores are much larger than the

response peak score (R_{max} = 7.126) as shown in Figure 2 (point A and point D), and the tracking results in these two frames are regarded to be highly reliable. Therefore, there is no need for activating detector in this case. However, when the target object underwent significant appearance changes, such as occlusion and deformation, the *PSR* scores are lower than the response peak score (from point B to point C). As you can see in Figure 3, the target is experiencing occlusion from #70 to #79 in Jogging-1. In order to ensure accurate tracking of the target in subsequent frames, we activate the detector timely to re-detect the target object and avoid tracking drift under the circumstance. In this work, we consider the response map peak to be a dynamic threshold in comparison to its corresponding *PSR* score. Only when the *PSR* score is lower than its corresponding response map peak score R_{max} and the tracking failure occurred at this point, is the detector activated online for re-detecting. Otherwise, the detector is not activated at this time.

In MOSSE [5], when the *PSR* drops below 10, the occlusion appears or the tracking fails. Therefore, we first set the fixed threshold to 10 and the other parameters were consistent. We select multiple video sequences in the OTB dataset for testing. The results are shown in Table 1. We consider the response map peak of each sequence to be a dynamic threshold, which is better than using a fixed threshold in most video sequences.

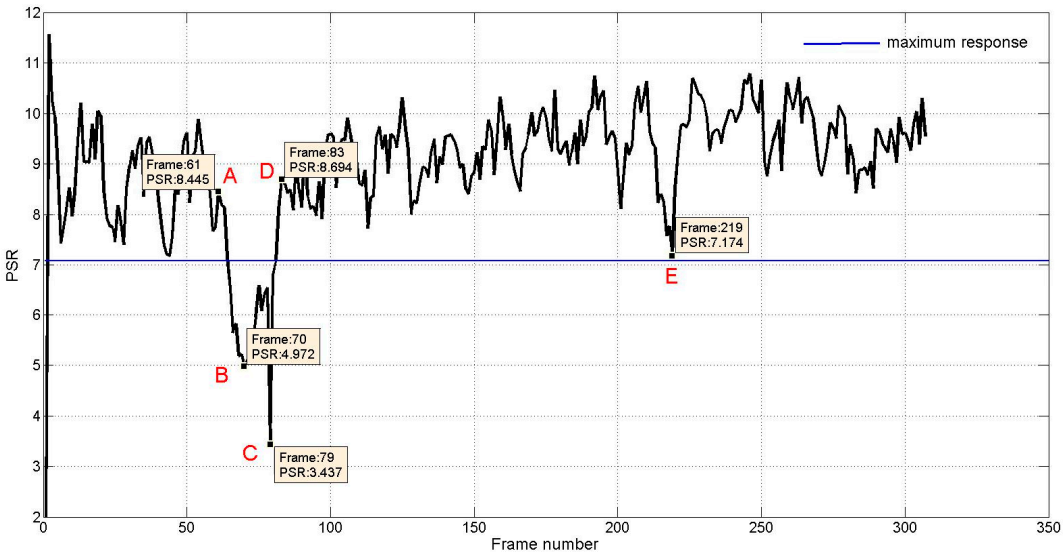


Figure 2. Peak-to-side lobe ratio (*PSR*) distribution.

Figure 3. Partial tracking result in *Jogging-1* sequence.

Table 1. Comparison of fixed threshold and dynamic threshold.

Precision	Crossing	Crowds	Couple	Surfer	Girl	Motor Rolling	David3
Fixed Threshold	0.983	0.896	0.929	0.753	1	0.878	1
Dynamic Threshold	1	0.991	0.929	0.843	1	0.957	1

3.4. Model Update

Many tracking methods now update their models frame-by-frame or at a fixed interval. However, there exists some disadvantage of these methods. When the target objects go through complicated

scenario changes such as occlusion, scale changes, and deformation, the tracking models may have absorbed some wrong information. The wrong information will be delivered to subsequent frames and they will decrease the performance of tracking after accumulating for a long time. The end result is tracking drift.

To solve the issues of model update, we propose a novel update method which is equivalent to active mode. In this paper, the response map peak is taken as a dynamical threshold. Since the corresponding response map peak score is different at each frame, our approach is to compare the size of *PSR* score and corresponding response map peak score. Only when the *PSR* score is greater than its maximum response peak score R_{max}, will this show that the tracking result has a good performance in the current frame. In this respect, the translation discriminative filter model (see Equation (2)) and the scale discriminative filter model (see Equation (8)) will be updated online based on a learning rate parameter η (see Equations (3) and (9)). Or else, the proposed approach chooses not to update the tracking model in the current frame. This can effectively prevent the error information from being passed to the next frame that cause tracking drift.

Figure 2 illustrates the distribution of the *PSR* scores. We can see from Figure 3 that the target objects go through complicated scenario challenges such as occlusion and background clutter from frames #61 to #79 (point A to point C), and the *PSR* score obviously decrease (that is, the *PSR* score decrease from 8.445 to 3.427). It is not suitable to update the model when the *PSR* score is lower than its corresponding peak score. When the target object left the occluded area at #83 (point D), the *PSR* score obviously increased (that is, the *PSR* score increase from 3.427 to 8.694). The greater the peak scores are, the more robust the tracking performance is. Under this circumstance, the update condition is met where the *PSR* score is greater than its corresponding peak score. The model update should be considered, and the tracking result is considered to be highly reliable in the current frame. In the case of point E (Figure 3), *PSR* score and its corresponding peak score has similar values. At this time, it is also not suitable to update the model.

3.5. Algorithm Flowchart

The proposed tracking algorithm is showed in Algorithm 1.

Algorithm 1. Proposed tracking algorithm

Input: Initial target location P_0 (x_0, y_0), initial scale S_0 (w_0, v_0), initial *PSR* score PSR_0, hierarchical correlation filters $\{W_t^l \mid l = 3, 4, 5\}$.
Output: Estimated object location P_t (x_t, y_t), estimated scale S_t (w_t, v_t).

Repeat:
Crop out the image samples centered at P (x_t, y_t) and extract convolutional features and HOG features;
For each layer l computes the response map $R_{context}$ via Equation (2);
Estimate the location of the target by computing the maximum response map after fusion R_t via Equation (6);
Construct a target scale pyramid around P (x_t, y_t) and estimate the optimal scale of the target as in Equation (10);
Calculate the *PSR* score of the response map peak;
If $PSR_t < R_{max}$, **then**
Activate re-detecting component D and find the possible candidate states C;
For each state C_d in C, do computing response score $R_{context}$ via Equation (2);
End if
If $PSR_t > R_{max}$, **then**
Update the tracking model via Equations (3a) and (3b);
Update the scale estimation model via Equations (9a) and (9b);
Update detector D;
End if
Until end of video sequence.

4. Experimental Results

4.1. Implementation Details

The proposed tracker is implemented in MATLAB2014a on a PC with an i7 3.2 GHz CPU with 16 GB memory. The feature extractor we employ is pre-trained VGGNet-16 trained on the ImageNet dataset. We also use HOG features in 4×4 window (a cell size of 4×4). We set the size of the search window to 2.2 times the size of the target object. The spatial bandwidth is set to $1/10$. The learning rate parameter η in Equations (3) and (9) is set to 0.025. The number of scales (S) is set to 33, with a scale-step of 1.02. The PSR_0 (the PSR initial value) is set to 1.8. We employ the same parameters for each video sequence.

The position of the accuracy degradation is different for different video sequences (Table 2).

Table 2. Different scale-step values and their corresponding precision.

Sequence	Scale-Step	Precision
Jogging-1	1.02	0.974
	1.03	0.974
	1.04	0.974
	1.05	0.974
	1.2	0.915
	1.3	0.863

4.2. The Overall Tracking Performance on OTB Benchmark Datasets

We evaluate the proposed tracker, and it compares with the state-of-the-art 13 trackers including C-COT_HOG [23], SRDCF [21], CFNet_conv3 [41], SiamFC_3s [42], STAPLE_CA [13], Staple [17], LMCF [22], SAMF_AT [43], SAMF_CA [13], LCT [12], DSST [16], KCF [6], and some CNN based tracker, including CNN-SVM [44] on a large standard benchmark dataset [3], [4] that contains 100 videos. We use three metrics provided in [3] and [4] to evaluate 11 trackers on OTB-100, and display the tracking results by two indicators (Distance Precision (DP) and Overlap success Precision (OP)). We evaluate the tracking performance based on one-pass evaluation (OPE) protocol provided by [3] and [4]. As shown in Figure 4, we use DP plot and OP plot for presentation and employ area under the curve (AUC) success plots to rank these trackers.

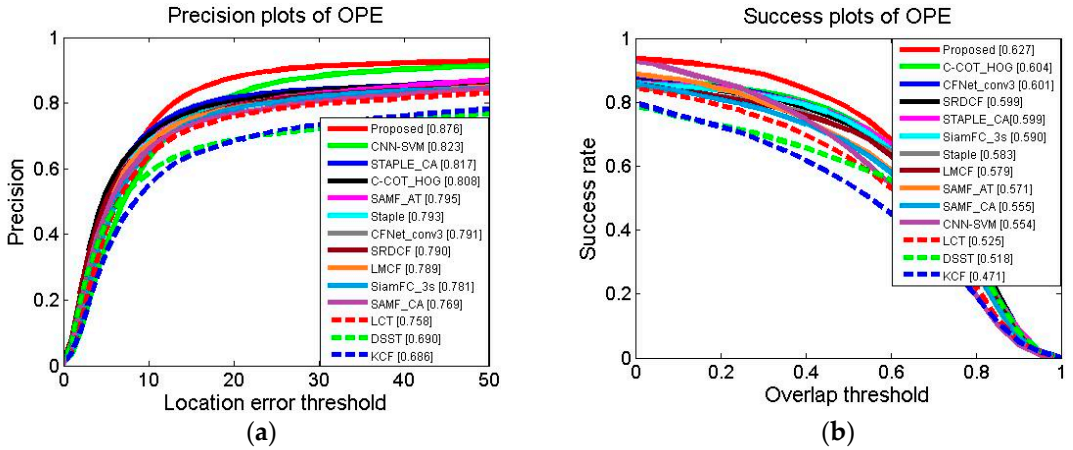


Figure 4. The overall tracking performances of precision (**a**) and success plots (**b**) comparing the proposed tracker with state-of-the-art trackers on OTB-100 using OPE evaluation. The proposed tracker performs well against these algorithms.

Figure 4 illustrates DP plot and OP plot of 12 trackers on OTB-100 benchmark datasets. As can be seen in the figure, the proposed tracker performs favorably against state-of-the-art trackers in distance precision (DP) and overlap success precision (OP). DP is computed as the relative number of frames in the sequence where the center location error is smaller than a certain threshold, and the DP values at a threshold of 20 pixels are reported. OP is defined as the percentage of frames where the bounding box overlap larger than a given threshold and the initial threshold value of overlap success (OP) is generally set to 0.5. Table 3 shows the results from the proposed tracker and the state-of-the-art trackers. The proposed tracker performs better with DP of 87.6% and OP of 62.7%, where the DP and OP surpassed other trackers. The comparison of these tracking results prove the effectiveness of the proposed methods. The last column of bold Proposed is the best result.

The proposed tracker mainly employs the hierarchical features of CNNs for feature representation and the tracking speed is 6 frames per second, the main time-consuming burden of the proposed tracker is the process of extracting deep convolutional features. The next work guarantees robustness while improving tracking speed.

Table 3. Comparison with state-of-the-art trackers on the OTB dataset. The results are presented in terms of distance precision (DP) and overlap success precision (OP).

	KCF	**DSST**	**LCT**	**CNN-SVM**	**SAMF_CA**	**SAMF_AT**	**LMCF**
DP	0.686	0.690	0.758	0.823	0.769	0.795	0.789
OP	0.471	0.518	0.525	0.554	0.555	0.571	0.579
	Staple	**SiamFC_3s**	**STAPLE_CA**	**SRDCF**	**CFNet_conv3**	**C-COT_HOG**	**Proposed**
DP	0.790	0.781	0.817	0.793	0.791	0.808	0.876
OP	0.583	0.590	0.599	0.599	0.601	0.604	0.627

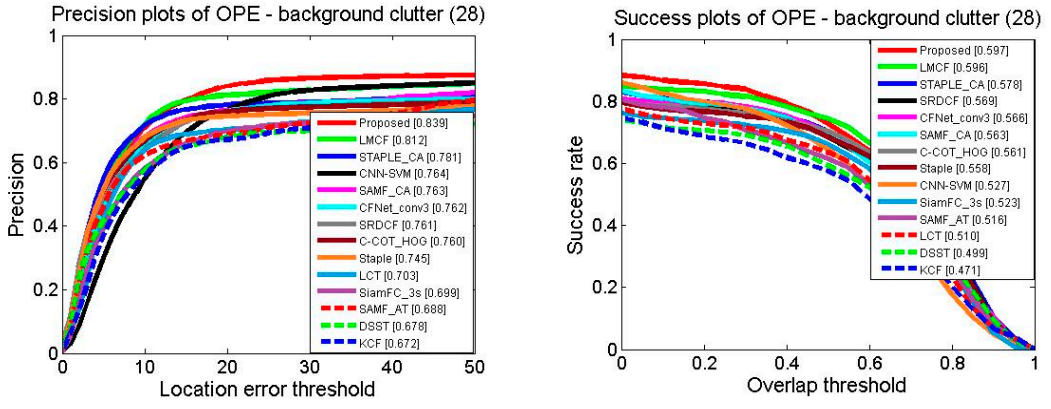
4.3. The Attribute-Based Tracking Evaluation

We also perform an attribute-based analysis of our approach. In the OTB dataset, all videos are annotated with 11 different attributes, namely: illumination variation, motion blur, fast motion, in-plane rotation, scale variation, background clutter, deformation, out of view, out-of-plane rotation, occlusion and low resolution. Due to the limited space, we only display 5 attributes results for representation in Figure 5.

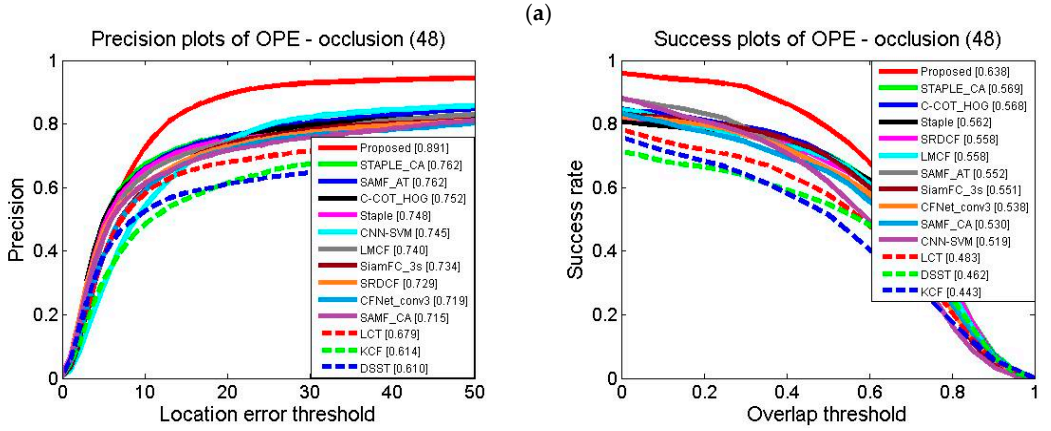
Figure 5 illustrates that the proposed tracker compares to 13 state-of-the-art trackers, and the result use DP plot and OP plot to show four different challenges attributes. It is clearly demonstrated that proposed tracker obtains superior DP and OP in background clutter (83.9%, 59.7%), occlusion (89.1%, 63.8%), in-plane rotation (86.2%, 61.0%), illumination variation (83.7%, 60.8%), and scale variation (86.6%, 60.2%). In sequences annotated with the background clutter attribute, fast motion attribute and in-plane rotation attribute, our approach outperforms the compared trackers. Benefit from the re-detecting method and self-adaptive model update method, the proposed tracker can adaptively activate the re-detection and update tracking model in the case of tracking failure. This shows that our tracker is highly robust and achieves superior performance in different challenging scenarios.

4.4. Qualitative Evaluation

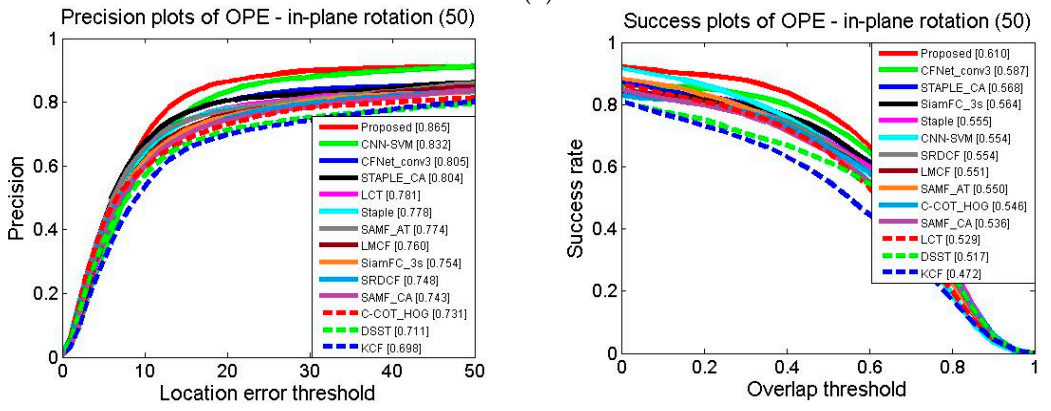
Here we provide a qualitative comparison result of our approach with six state-of-the-art trackers from the literature (C-COT_HOG, SRDCF, STAPLE_CA, LMCF, SAMF_AT, and CNN-SVM). Figure 6 illustrates frames from five sequences with scale variation (Dragon Baby, Ironman, Soccer, Box), occlusion (Dragon Baby, Ironman, Soccer, Box, Bird2), background clutters (Ironman, Soccer, Box), motion blur (Dragon Baby, Ironman, Soccer, Box), in-plane rotation (Dragon Baby, Ironman, Soccer, Box, Bird2), out-of-plane rotation (Dragon Baby, Ironman, Soccer, Box, Bird2), illumination variation (Ironman, Soccer, Box).



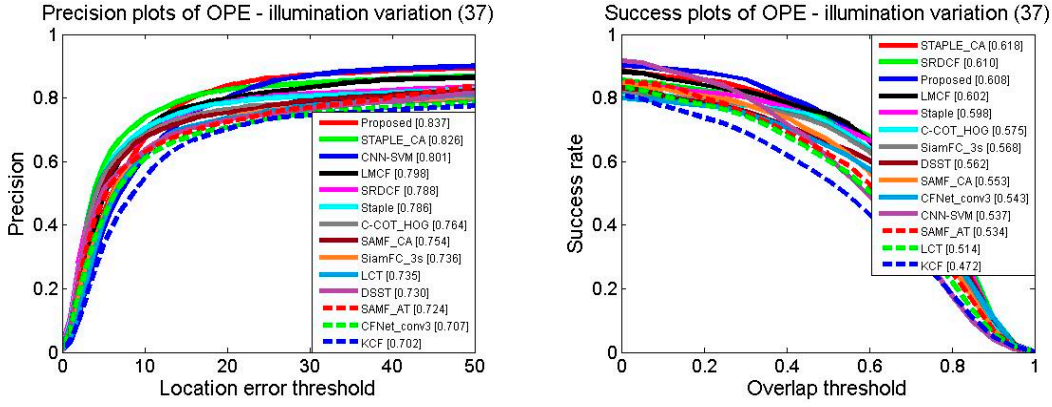
(**a**)



(**b**)

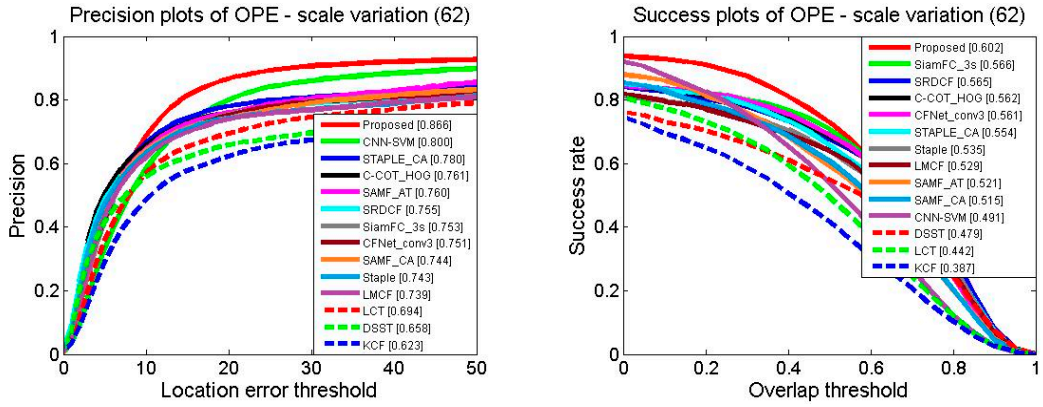


(**c**)



(**d**)

Figure 5. *Cont*.



(**e**)

Figure 5. Attribute-based evaluation of the distance precision and overlap success plots compare the proposed algorithm with 13 state-of-the-art trackers over four video attributes of (**a**) background clutter, (**b**) occlusion, (**c**) in-plane rotation, (**d**) illumination variation, and (**e**) scale variation.

In the Bird2 and Dragon Baby sequences, the compared trackers struggle due to occlusion, in-plane rotation and out-of-plane rotation, while our tracker robustly handles these factors. In the Ironman, Soccer and Box sequences, the target mainly goes through complicated scenarios challenges, for instance scale variation, motion blur, and background clutters. Most of the trackers such as C-COT_HOG, SRDCF, STAPLE_CA, SAMF_AT and CNN-SVM trackers lose the target and fail to recover. Benefit from the re-detection activation strategy and the self-adaptive model updates strategy, the proposed tracker can track the target successfully, which further validate the effectiveness of the proposed methods.

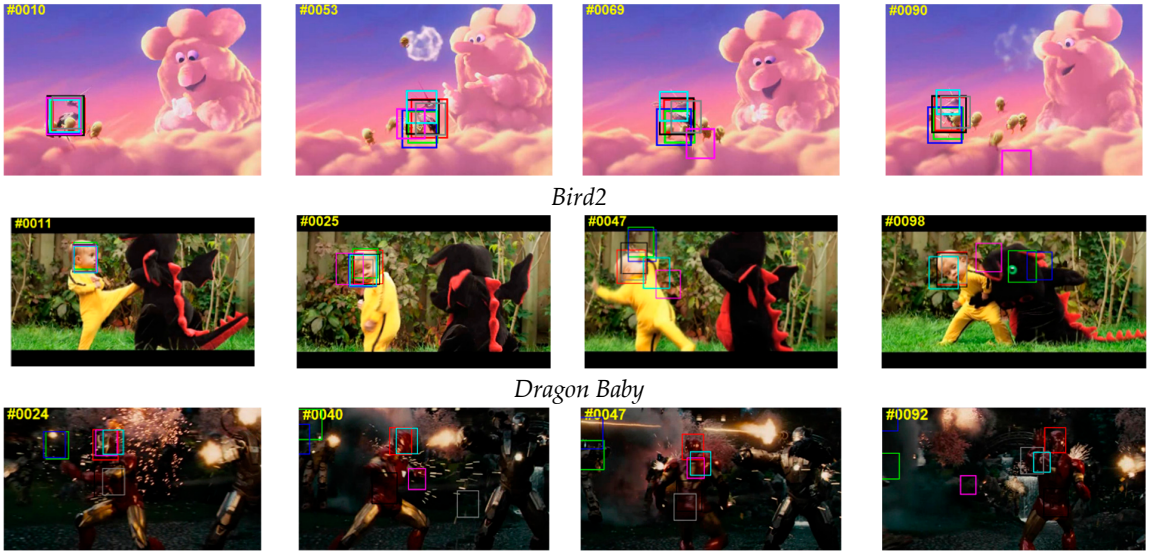


Figure 6. *Cont.*

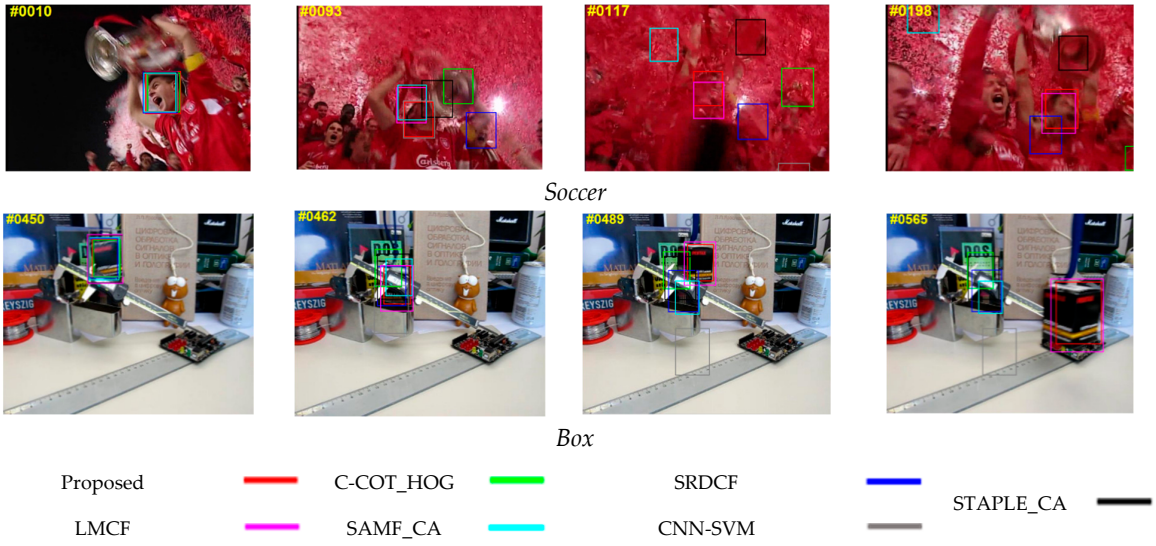


Figure 6. A qualitative comparison of our approach with six trackers, C-COT_HOG [23], SRDCF [21], STAPLE_CA [13], LMCF [22], SAMF_CA [13], CNN-SVM [44]. Tracking results are shown on five video sequences from the OTB dataset (from top to down are *Bird2*, *Dragon Baby*, *Ironman*, *Soccer*, *Box*). Our approach performs favorably compared to the existing tracker in these challenging situations.

5. Conclusions

In this paper, we use hierarchical features of CNNs for visual tracking. We make full use of the characteristics of each convolution layer to locate the object target. The semantic information of deep convolution has great significance for the change of the appearance of the target. The spatial details of shallow convolutional layer enable to locate targets position accurately. In addition, we train a spatial-temporal context filter on convolutional layers and predict the target position by fusing the response value of the filters on the three convolutional layers. Moreover, we proposed a re-detection method and model update method by comparing the size of the *PSR* and its corresponding response value to determine whether to re-detect or update the model. The experimental results demonstrate that the proposed algorithm with deep and HOG features outperforms most state-of-the-art algorithms based on DCFs.

Author Contributions: Writing—original draft, W.Z.; Writing—review & editing, Y.L., Z.C., Y.D., D.Z. and P.L.

Funding: This work was supported by the Quanzhou scientific and technological planning projects, Fujian, China (No.2018C113R), and the grants from National Natural Science Foundation of China (Grant No.61605048), in part by Natural Science Foundation of Fujian Province, China under Grant 2016J01300, in part by Fujian Provincial Big Data Research Institute of Intelligent Manufacturing, Quanzhou, 362021, China, in part by Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No.ZQN-PY518), and in part by the Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University under Grant 17014084014.

Acknowledgments: Thanks to my family for their support. Thanks to my teachers and classmates for giving me guidance on my studies. Thanks to my school Huaqiao University.

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

References

1. Smeulders, A.W.; 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]
2. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. [CrossRef]
3. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
4. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]

5. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
6. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–5 December 2012; pp. 1097–1105.
8. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representation. *arXiv* **2014**, arXiv:1409.1556.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. 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, Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
11. Zhang, K.; Zhang, L.; Yang, M.H.; Zhang, D. Fast tracking via spatio-temporal context learning. *arXiv* **2013**, arXiv:1311.1939.
12. Ma, C.; Yang, X.; Zhang, C.; Yang, M.H. Long-term correlation tracking. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
13. Mueller, M.; Smith, N.; Ghanem, B. Context-Aware Correlation Filter Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1387–1395.
14. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
15. Zitnick, C.L.; Dollár, P. Edge Boxes: Locating Object Proposals from Edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
16. Danelljan, M.; Häger, G.; Khan, F.S. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; pp. 65.1–65.11.
17. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
18. 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 Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4266–4274.
19. Rui, C.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715.
20. Zhang, K.; Zhang, L.; Liu, Q.; Zhang, D.; Yang, M.H. Fast Visual Tracking via Dense Spatio-temporal Context Learning. In *ECCV 2014 Computer Vision—ECCV 2014*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8693, pp. 127–141.
21. 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, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
22. Wang, M.; Liu, Y.; Huang, Z. Large Margin Object Tracking with Circulant Feature Maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4800–4808.
23. 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, Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
24. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
25. Alam, M.S.; Ochilov, S. Spectral fringe-adjusted joint transform correlation. *Appl. Opt.* **2010**, *49*, B18–B25. [CrossRef] [PubMed]

26. Sidike, P.; Asari, V.K.; Alam, M.S. Multiclass Object Detection with Single Query in Hyperspectral Imagery Using Class-Associative Spectral Fringe-Adjusted Joint Transform Correlation. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1196–1208. [CrossRef]
27. Krieger, E.; Aspiras, T.; Asari, V.K.; Krucki, K.; Wauligman, B.; Diskin, Y.; Salva, K. Vehicle tracking in full motion video using the progressively expanded neural network (PENNet) tracker. *Proc. SPIE* **2018**, *10649*. [CrossRef]
28. Krieger, E.W.; Sidike, P.; Aspiras, T.; Asari, V.K. Deterministic object tracking using Gaussian ringlet and directional edge features. *Opt. Laser Technol.* **2017**, *95*, 133–146. [CrossRef]
29. Zhang, M.; Wang, Q.; Xing, J.; Gao, J.; Peng, P.; Hu, W.; Maybank, S. Visual Tracking via Spatially Aligned Correlation Filters Network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 469–485.
30. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.H.; Yang, M.H. VITAL: VIsual Tracking via Adversarial Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
31. 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 Workshop, Santiago, Chile, 7–13 December 2015; pp. 621–629.
32. Fan, J.; Xu, W.; Wu, Y.; Gong, Y. Human tracking using convolutional neural networks. *IEEE Trans. Neural Netw.* **2010**, *21*, 1610–1623. [PubMed]
33. Li, H.; Li, Y.; Porikli, F. DeepTrack: Learning Discriminative Feature Representations by Convolutional Neural Networks for Visual Tracking. *IEEE Trans. Image Process.* **2014**, *25*, 1834–1848. [CrossRef] [PubMed]
34. Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.H. Robust Visual Tracking via Convolutional Networks Without Training. *IEEE Trans. Image Process.* **2016**, *25*, 1779–1792. [CrossRef] [PubMed]
35. 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 IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2574–2583.
36. Zhu, Z.; Huang, G.; Zou, W.; Du, D.; Huang, C. UCT: Learning Unified Convolutional Networks for Real-Time Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Venice, Italy, 22–29 October 2017; pp. 1973–1982.
37. Yao, Y.; Wu, X.; Zhang, L.; Shan, S.; Zuo, W. Joint Representation and Truncated Inference Learning for Correlation Filter based Tracking. *arXiv* **2018**, arXiv:1807.11071.
38. Zhang, T.; Xu, C.; Yang, M.H. Multi-task Correlation Particle Filter for Robust Object Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4819–4827.
39. Wang, L.; Liu, T.; Wang, G.; Chan, K.L.; Yang, Q. Video tracking using learned hierarchical features. *IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc.* **2015**, *24*, 1424–1435. [CrossRef] [PubMed]
40. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 447–456.
41. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
42. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In *Computer Vision—ECCV 2016 Workshops. ECCV 2016*; Hua, G., Jégou, H., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9914.
43. Bibi, A.; Mueller, M.; Ghanem, B. Target Response Adaptation for Correlation Filter Tracking. In *Computer Vision—ECCV 2016*; Springer: Cham, Switzerland, 2016; pp. 419–433.
44. 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 International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 597–606.