

Article

# Adaptive Correlation Model for Visual Tracking Using Keypoints Matching and Deep Convolutional Feature

Yuankun Li <sup>1</sup>, Tingfa Xu <sup>1,2,\*</sup>, Honggao Deng <sup>1,3</sup>, Guokai Shi <sup>1</sup> and Jie Guo <sup>1</sup> 

<sup>1</sup> School of Optics and Photonics, Image Engineering & Video Technology Lab, Beijing Institute of Technology, Beijing 100081, China; liyuankunbixian@gmail.com (Y.L.); dhg007@sina.com (H.D.); shi\_guokai\_123@126.com (G.S.); jieguo\_2013@163.com (J.G.)

<sup>2</sup> Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China, Beijing 100081, China

<sup>3</sup> School of Information and Communication, Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China

\* Correspondence: ciom\_xtf1@bit.edu.cn; Tel.: +86-10-6891-2567

Received: 7 November 2017; Accepted: 7 February 2018; Published: 23 February 2018

**Abstract:** Although correlation filter (CF)-based visual tracking algorithms have achieved appealing results, there are still some problems to be solved. When the target object goes through long-term occlusions or scale variation, the correlation model used in existing CF-based algorithms will inevitably learn some non-target information or partial-target information. In order to avoid model contamination and enhance the adaptability of model updating, we introduce the keypoints matching strategy and adjust the model learning rate dynamically according to the matching score. Moreover, the proposed approach extracts convolutional features from a deep convolutional neural network (DCNN) to accurately estimate the position and scale of the target. Experimental results demonstrate that the proposed tracker has achieved satisfactory performance in a wide range of challenging tracking scenarios.

**Keywords:** correlation filter-based visual tracking; deep convolutional neural network; deep convolutional feature; keypoints matching; adaptive model updating

## 1. Introduction

As one of the fundamental research topics in military, security, and human–computer interaction, visual tracking plays an important role in many applications. Given the initial motion state of the target object in the first frame, a visual tracking algorithm aims to estimate the motion state of the target object in each subsequent frame. Despite the massive work done in recent years, visual tracking is still challenging due to the appearance of variations caused by occlusion, target rotation, scale variation, and so on.

The correlation filter (CF) was originally designed to generate correlation peak output for an input signal. According to the convolution theorem, correlation operations can be significantly accelerated using fast Fourier transformation. In general, CF-based visual trackers use correlation filters to model the appearance of the target and update correlation filters at each frame using a fixed learning rate. However, due to the unreliable tracking caused by occlusion, background clutter, and other perturbations, correlation models may be contaminated during the updating process. In order to alleviate model contamination, dynamic adjustment of learning rate will be necessary and constructive.

To account for the target appearance changes over time, man kinds of feature descriptors have been used in visual tracking, such as Haar-like features [1], Color Names [2], FAST [3], and HOG [4].

Recently, features learned from deep convolutional neural networks (DCNNs) have been used in a variety of visual tasks. However, the outstanding performance of DCNNs relies heavily on training on large-scale datasets. Thus, the application of DCNNs in visual tracking is severely restricted by the very limited information in the first frame. Several tracking approaches [5,6] use pre-trained DCNNs as feature extractors. However, these approaches use only a few convolutional layers and cannot provide a comprehensive description of the target state.

Based on the discussion above, we propose a visual tracking framework that utilizes DCNN and CF synthetically. The main contributions of this paper are as follows:

1. We propose a novel model updating method. Firstly, we establish a keypoints library to restore the reliable historical data, and then we obtain the pixel-level correspondence between the current frame and the previous frame using dense matching. Finally, the similarity score is calculated by comparing matched pairs of keypoints and is used to adjust the learning rate in model updating.
2. We propose a method to fully exploit the hierarchical features generated of the DCNN, which can make full use of spatial detail information and semantic information.
3. Based on the observation of different layers' output, we propose a scale estimation method using deep convolutional features.

The rest of the paper is organized as follows: In Section 2, we review research work related to ours. In Section 3, we present the proposed visual tracking framework in detail. Numerous experimental results and analysis are shown in Section 4. In Section 5, we reach the conclusions of our work.

## 2. Related Work

In this section, we list some works closely related to ours.

### 2.1. Trackers with Convolutional Neural Network

In recent years, convolutional neural networks (CNNs) have made significant progress on a wide range of computer vision issues, including visual tracking. Based on the combination of off-line pre-training and on-line fine-tuning, Wang et al. proposed the deep learning tracker (DLT)[7] and the structured output deep learning tracker (SO-DLT) [8] in the framework of particle filters. To avoid the issues caused by offline training, trackers in [9,10] incrementally learn target-specific CNNs without pre-training. The trackers mentioned above simply treat the CNN as a black-box classifier where only the outputs of the last layer are used to represent the target object. However, the goal of visual tracking is to estimate the target state precisely rather than to infer their semantic classes. Ma et al. [5] extract the hierarchical convolutional features (HCF) from three layers of CNN to learn multiple correlation filters for visual tracking. Danelljan et al. [11] proposed a tracker by learning continuous convolution operators (CCOT) to interpolate discrete features and train spatial continuous convolution filters, which enables efficient integration of multi-resolution deep feature maps. To alleviate the low computational efficiency caused by CNN operation, Danelljan et al.[12] designed an efficient convolution operators (ECO) for visual tracking using a factorized convolution operation. Although CCOT and ECO trackers use convolutional features for translation estimation, neither of them takes full advantage of the entire CNN and thus lack a complete description of the motion state of the target. Besides, compared with HCF tracker, CCOT and ECO trackers focus on the improvements to the CF model and ignore the problems during the model updating process.

### 2.2. Trackers with Correlation Filters

Since Bolme et al. [13] introduced correlation filters into visual tracking by minimizing the output sum of squared error (MOSSE), CF-based visual tracking algorithms have attracted considerable attention due to their high speed. Based on the raw pixel data, the MOSSE tracker performs high-speed CF training and tracking. The circulant structure and kernelized operator (CSK) [14] introduced in CF-based visual tracking algorithm significantly improves the capacity of the training set and

thus improves the tracking accuracy. CSK was then extended to [2] and [4] by leveraging the HOG feature and the Color Names feature, respectively. Danelljan et al. [15] designed a scale estimation correlation filter to predict the spatial size of the target. In order to alleviate the boundary effect caused by circulant structure, Danelljan et al. [16] introduced spatial regularization in the cost function of correlation filters. By introducing mask matrix and sample cropping, Galoogahi et al. [16] alleviated the boundary effect in a different way. Based on the similarity between correlation and convolution operations, Valmadre et al. [17] construct a CNN where the correlation filter is part of the network and achieve end-to-end representation learning.

### 2.3. Trackers with Keypoints and Matching

Part-based visual tracking methods have exhibited outstanding performance against occlusion. While some trackers [18,19] choose rectangular parts as matching parts, the size and number of rectangular parts limit the speed of tracking. Instead, matching with feature point descriptors (such as SIFT [20] and BRISK [21]) is rather computationally convenient, which makes keypoints an ideal representation for modeling local part. In [22], Grabner et al. employed a boost classifier to obtain keypoints matching. Hare et al. [23] attached weights on different keypoints and update the weights in a unified framework. Tracker in [3] jointly uses optical flow tracking and keypoint matching to provide an estimate of both target position and target rotation.

## 3. Proposed Approach

Figure 1 shows an overall flow of the proposed ACMD (Adaptive Correlation model for visual tracking using keypoints Matching and Deep convolutional feature) tracker. An input image of the  $t$ -th frame is first pre-processed to fit the network input. Then, the conv2-2, conv3-4, conv4-4, and conv5-4 layers of VGG-19 [24] are used as feature extractors. These features are then convolved with two learned CF models to provide translation estimation and scale estimation. Final estimation of the  $t$ -th frame is achieved by the combination of two CF models' output. A dense matching is then employed between current frame and previous frame and the matching score is used to update the keypoints set and discount the learning rate.

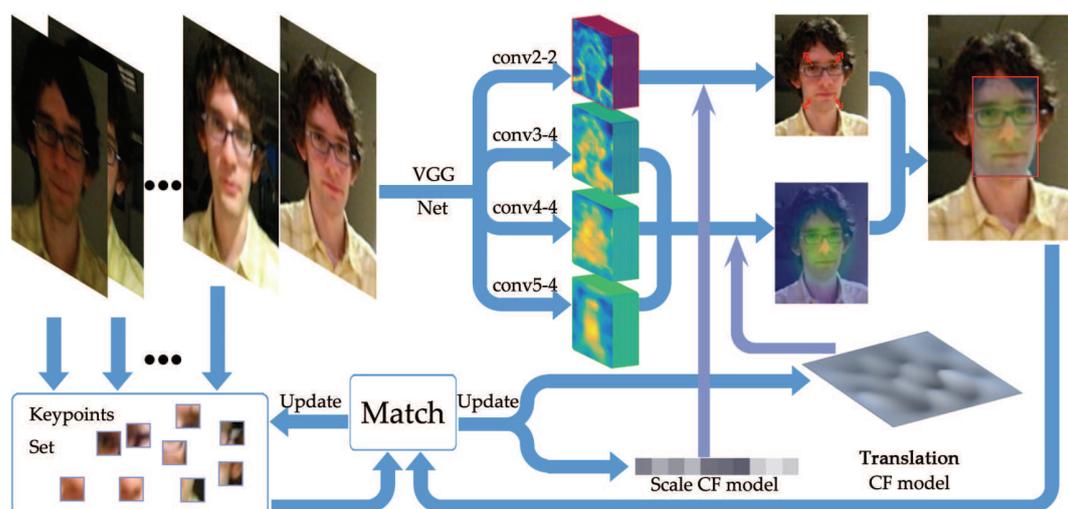
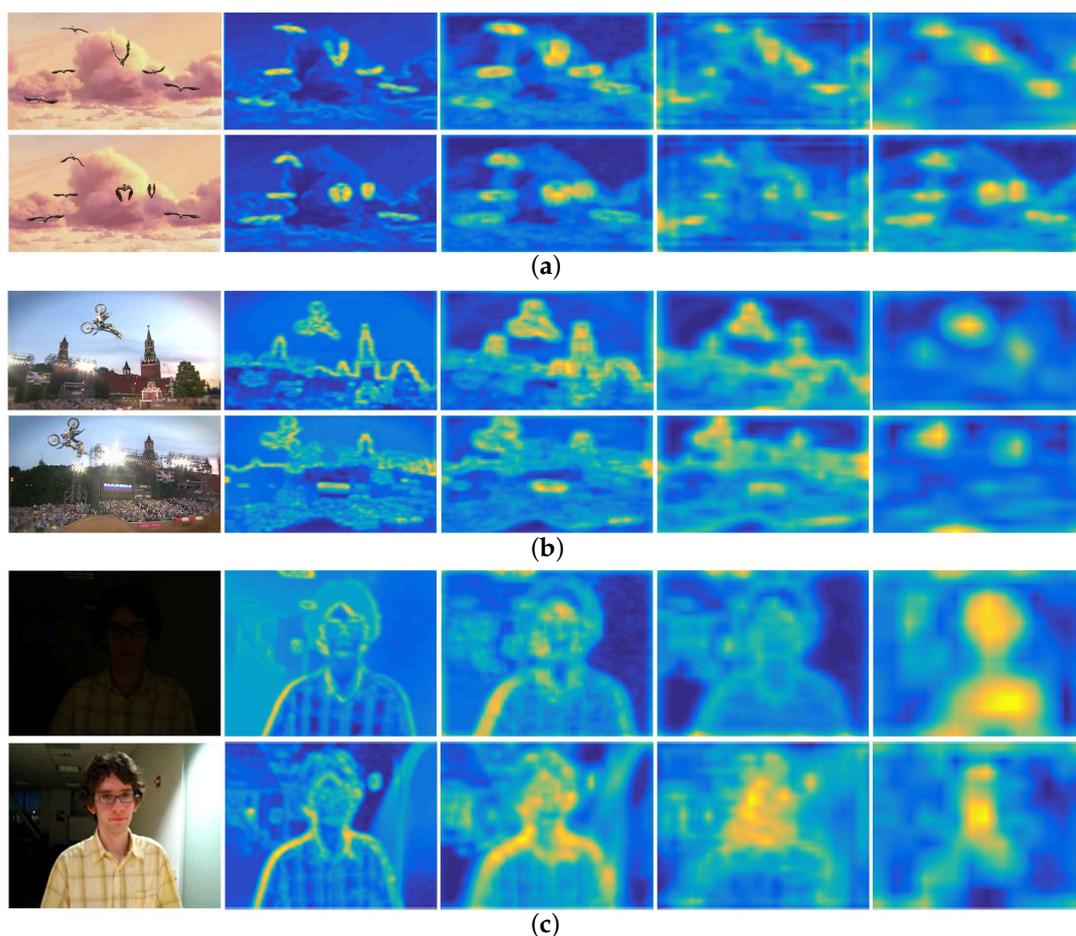


Figure 1. Flow chart of the proposed framework. CF: correlation filter.

### 3.1. Deep Convolutional Features

It should be noted that research [25] suggests that CNN's improved performance is obtained using convolutional layers rather than fully-connected layers. Hence, we use convolutional layers to extract features. Recent studies [5,11,25] and our experimental results illustrate that:

1. CNN feature maps are high-dimensional features and contain information highly related to the target state.
2. Different layers of CNN encode different types of information. Feature maps of higher layers encode semantic information. As shown in Figure 2, although the appearance of the targets undergoes different variation (non-rigid deformation in Bird1, in-plane-rotation in MotorRolling, illumination variation in David), the region around the target is always bright yellow in feature maps of conv5-4 layer. This character is quite useful when the target undergoes severe appearance variation.
3. Feature maps of lower layers retain more spatial details of the target, such as borders, corners, and curves. Taking David as an example, it is obvious that the texture of the face such as edges and contours are well preserved, including the corner of the ear, the boundary of the face, etc., which could be used to determine the boundary of the target and thus to make scale estimation.



**Figure 2.** Visualization of input and outputs of different layers. From left to right are the input frame, feature maps of conv2-2, feature maps of conv3-4, feature maps of conv4-4, and feature maps of conv5-4. (a) Bird1; (b) MotorRolling; (c) David.

The net we employed is the VGG-19 [24] network, which was pre-trained offline using the ImageNet [26] dataset for classification tasks. It should be noted that the VGG-19 network takes

224 × 224 RGB images as input, so the input frame must be resized first. Additionally, on account of the pooling method, the spatial resolution decreases gradually as the net propagates forward, so an upsampling process is necessary after we extract raw convolutional features. We apply bilinear interpolation for both resizing and upsampling processes.

Let  $x$  denote the input RGB image patch and  $f_c$  denote the feature map generated by the  $c$ -th convolutional layer. After the feature extraction procedure,  $f_c$  shares the same spatial resolution with  $x$ , while the dimensionality of  $f_c$  is determined by the value of  $c$ .

### 3.2. Correlation Filter

Traditionally, the goal of training is to find a correlation filter template  $h$  that minimizes the output of Equation (1):

$$\min_h \|h^T f - g\|^2 + \lambda \|h\|^2. \quad (1)$$

Equation (1) is the form of the cost function of ridge regression, in which  $f$  is the training sample,  $g$  is the desired output, and  $\lambda \geq 0$  is the regularization parameter. The superscript  $T$  denotes matrix transpose operation.

#### 3.2.1. Correlation Filters for Translation Estimation

It should be noted that only one training sample in one dimensionality is taken into consideration in Equation (1). A circulant structure is used to generate a set of training samples  $\{f_{m,n} | (m,n) \in \{0,1,\dots,M\} \times \{0,1,\dots,N\}\}$ . Equation (1) can be transformed into the following form:

$$\min_h \sum_{m,n} \left\| \sum_{l=1}^L (h_{m,n}^l)^T f_{m,n}^l - g(m,n) \right\|^2 + \lambda \sum_{l=1}^L \|h^l\|^2, \quad (2)$$

where superscript  $l$  denotes the  $l$ -th dimensionality of a matrix and  $g(m,n)$  is the GAUSSIAN shaped label:

$$g(m,n) = \exp \left[ -\frac{(m - M/2)^2 + (n - N/2)^2}{2\sigma^2} \right]. \quad (3)$$

According to [4], the solution to Equation (2) is:

$$H^l = \frac{G \star \overline{F^l}}{\sum_{i=1}^L F^i \star \overline{F^i} + \lambda}, \quad (4)$$

where capital letters denote the Fourier transformation form, the overbar notation denotes complex conjugation form, and the  $\star$  operator performs an element-wise multiplication of the two matrices.

Given a sample patch  $f_c$ , the correlation response output  $y_c$  is calculated by Equation (5):

$$y_c = \mathcal{F}^{-1} \left( \sum_{l=1}^L H^l \star \overline{f_c^l} \right), \quad (5)$$

where  $\mathcal{F}^{-1}$  denotes the inverse Fourier transform operator. The final response output for translation estimation is obtained by a weighted average of all  $y_c$ :

$$y_{trans} = \frac{\sum_c (\mu_c y_c)}{\sum_c \mu_c}. \quad (6)$$

The new target center is estimated to be at the position of maximum value of  $y_{trans}$ .

### 3.2.2. Correlation Filters for Scale Estimation

The correlation filter for scale estimation can also be obtained by the minimization Equation (1). Note that in scale space, samples expand in one dimension, so the set of training samples is generated like  $\{f_s^l(s) \in \{0, 1 \dots, S\}\}$ , and label in GAUSSIAN shaped  $g(s)$  is as follows:

$$g(s) = \exp\left[-\frac{(s - S/2)^2}{2\sigma^2}\right]. \quad (7)$$

Equation (1) can be rewritten as follows:

$$\min_h \sum_s \left\| \sum_{l=1}^L (h_s^l)^T f_s^l - g(s) \right\|^2 + \lambda \sum_{l=1}^L \|h_s^l\|^2. \quad (8)$$

The solution to Equation (8) is same as Equation (4), and the correlation response output can be calculated by:

$$y_{scale} = \mathcal{F}^{-1} \left( \sum_{l=1}^L H^l \star \overline{f_s^l} \right). \quad (9)$$

The current scale shares the same coordinate index with the maximum value of  $y_{scale}$ .

### 3.3. Adaptive Model Updating

The correlation model learned at the  $t$ -th frame  $H_t^l$  can be expressed in fractional form. Conventionally, the numerator  $A_t^l$  and denominator  $B_t$  of the correlation filter are updated using fixed learning rate  $\eta$ :

$$A_t^l = (1 - \eta_t) A_{t-1}^l + \eta_t (\overline{G}_t \star F_t^l), \quad (10)$$

$$B_t = (1 - \eta_t) B_{t-1} + \eta_t \left( \sum_{i=1}^L F^i \star \overline{F^i} \right). \quad (11)$$

It is not difficult to conclude that when the  $T$ -th frame arrives, the contribution rate of the  $t$ -th frame ( $1 < t < T$ ) to the model is:  $(1 - \eta)^{T-t} \eta$ . This can lead to two problems. First, the unreliable tracking data of the  $t$ -th frame will affect the tracking results of all subsequent frames. Second, when long-term occlusion occurs, continuous learning of corrupt data will force the correlation model to fit the occlusion information and thus reduce the discrimination of the model. To alleviate these two problems, we propose an adaptive model updating method.

Let  $K_{t-1}$  denote the keypoints set established in the 1st  $\sim (t-1)$ -th frames and  $D_{t-1}$  denote the corresponding set of descriptors of  $K_{t-1}$ . We initialize  $K_1$  and  $D_1$  at the first frame using the FAST [27] detector and the BRISK [21] descriptor. Firstly, the pixel-level correspondence between current  $t$ -th frames and the  $t-1$ -th frame is obtained using dense matching; that is, finding the matched point  $k_{curr}$  in the current frame for  $k_{t-1} \in K_{t-1}$ .

$$k_{curr} = \mathcal{M}(k_{t-1}), \quad (12)$$

$$d_{curr} = \mathcal{D}(k_{curr}). \quad (13)$$

where  $\mathcal{M}$  is the dense matching process and  $\mathcal{D}$  is the descriptor calculation process. We use the Hamming distance to define the similarity between  $k_{curr}$  and  $k_{t-1}$ :

$$\mathcal{V}(k_{t-1}, k_{curr}) = 1 - \frac{\sum_i k_{t-1}[i] \oplus d_{curr}[i]}{v_{max}}. \quad (14)$$

where  $i$  is the index of sub-element in the descriptor and  $v_{max}$  is the maximum Hamming distance.  $\oplus$  denotes the exclusive-OR operation. In our case,  $v_{max}$  is equal to 512. Let  $K_{t-1}^{sub}$  denote the set of points in the  $K_{t-1}$  that participate in the dense matching process,  $K_{t-1}^{sub} \subset K_{t-1}$ . Additionally, let  $K_{curr}$  denote the set of points matched in the  $t$ -th frame. Then, the matching similarity score is calculated as follows:

$$\rho_t = \frac{1}{N} \sum \mathcal{V}(K_{t-1}, K_{curr}). \quad (15)$$

where  $\mathcal{V}(K_{t-1}, K_{curr})$  denotes the similarities of each element in set  $K_{t-1}$  and the corresponding element in set  $K_{curr}$ .

The learning rate  $\eta_t$  consists of two parts: the basic learning rate  $\eta_0$  and the discounting factor  $\rho_t$ :

$$\eta_t = \eta_0 \rho_t. \quad (16)$$

Since both Equation (14) and Equation (15) are normalized, the matching similarity score in Equation (15) can be used as the discount factor directly.

The feature point library can be updated as follows: for  $k_{curr}$ , if the similarity  $\mathcal{V}(k_{t-1}, k_{curr})$  between  $k_{curr}$  and  $k_{t-1}$  is larger than a threshold, use  $u_{curr}$  as  $u_t$ ; otherwise, use  $u_{t-1}$  as  $u_t$ .

$$\{k_t, d_t\} = \begin{cases} \{k_{curr}, \mathcal{D}(k_{curr})\}, \mathcal{V}(k_{t-1}, k_{curr}) \geq v_{thresh} \\ \{k_{t-1}, d_{t-1}\}, \mathcal{V}(k_{t-1}, k_{curr}) < v_{thresh} \end{cases} \quad (17)$$

The overall tracking algorithm is described in Algorithm 1.

---

**Algorithm 1:** Proposed tracking algorithm.

---

**Input :** Image  $I$ ; initial target position  $p_0$  and scale  $s_0$ ; previous target position  $p_{t-1}$  and scale  $s_{t-1}$ .

**Output :** Estimated object position  $p_t$  and scale  $s_t$ .

---

Initialize correlation filters  $H_1^{trans}, H_1^{scale}$  and set  $K_1, D_1$

**Foreach**  $I_t$

    Extract multiple  $f_c$  generated by VGG-Net;

    Compute the translation correlation  $y_{trans}$  using Equation (5) and Equation (6)

    Set  $p_t$  to at the maximum of  $y_{trans}$

    Compute the translation correlation  $y_{trans}$  using Equation (9)

    Set  $s_t$  to at the maximum of  $y_{scale}$

    Compute discounting factor  $\rho_t$  using Equation (12) to Equation (15)

    Update  $A_{trans}, B_{trans}, A_{scale}, B_{scale}, K, D$

**End**

---

## 4. Experiments

We evaluated the performance of our approach on the OTB-2013 [28] dataset and the OTB-2015 [29] dataset and compared our approach with 10 state-of-the-art trackers, including HCF [5], SRDCFad [30], SCT [31], MEEM [32], SAMF [33], DSST [15], KCF [4], STRUCT [34], TLD [35], SCM [36], and DLT [7]. Among those trackers, HCF and DLT use DCNN; HCF, SRDCFad, SCT, SAMF, DSST, and KCF are CF-based trackers and SRDCFad proposes an adaptive model updating method.

We implemented the proposed tracker in MATLAB 2015b. All of the experiments were performed on a PC with an Intel i7-4790 CPU. The speed of all trackers is shown in Table 1.

**Table 1.** Speed of the trackers.

	Ours	SRDCFad	HCF	SCT	MEEM	SAMF	DSST	KCF	STRUCT	TLD	SCM	DLT
Average FPS	3	3	6	44	11	12	56	192	10	22	0.4	0.6

The parameters, which are fixed for each sequence, are summarized as follows. The net employed for feature extraction was a pretrained version of VGG-19 [24]. We extracted the feature maps from the conv2-2, conv3-4, conv4-4, and conv5-4 layers. The basic learning rate  $\eta_0$  in Equation (16) was set to 0.025. To make scale estimation, we set  $S = 15$  and set the scale stride to 1.04. The weight in Equation (6) was set to  $\mu_{5_4} = 0.44, \mu_{4_4} = 0.33, \mu_{3_4} = 0.23$ . The dimensionality of keypoints descriptor was set to 512, and the threshold in Equation (17) was set to 300.

4.1. Quantitative Evaluation

We used the precision and success rate as the evaluation criteria of quantitative analysis. The precision criteria measure Euclidean distance between the center of tracker’s output and the ground truth. Precision plots show the percentage of frames whose precision is greater than a threshold. According to Reference [28], we used a threshold of 20 pixels. Another criteria—success rate—measures the coverage between tracker’s output and the ground truth. Assuming that the region of the tracker’s output is  $\gamma_t$  and the region of ground truth is  $\gamma_a$ , the success rate is defined as:

$$SuccessRate = \frac{|\gamma_t \cap \gamma_a|}{|\gamma_t \cup \gamma_a|}, \tag{18}$$

where  $\cap$  and  $\cup$  denote the intersection and union of two regions, respectively, and  $|\cdot|$  denotes the number of pixels in the region. The success plot illustrates the percentage of frames whose success rates are greater than a certain value. According to [28], we ranked all trackers using area under the curve (AUC) for the success rate.

Figures 3 and 4 illustrate the overall performance of all trackers in terms of the mentioned criteria. The proposed approach ranks first over both OTB-2013 and OTB-2015.

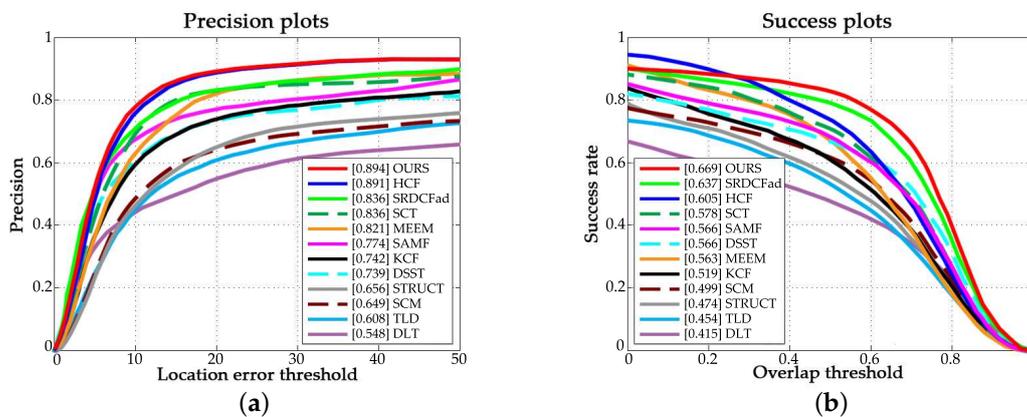


Figure 3. Precision and success plots over the OTB-2013 dataset. (a) Precision plot; (b) Success plots.

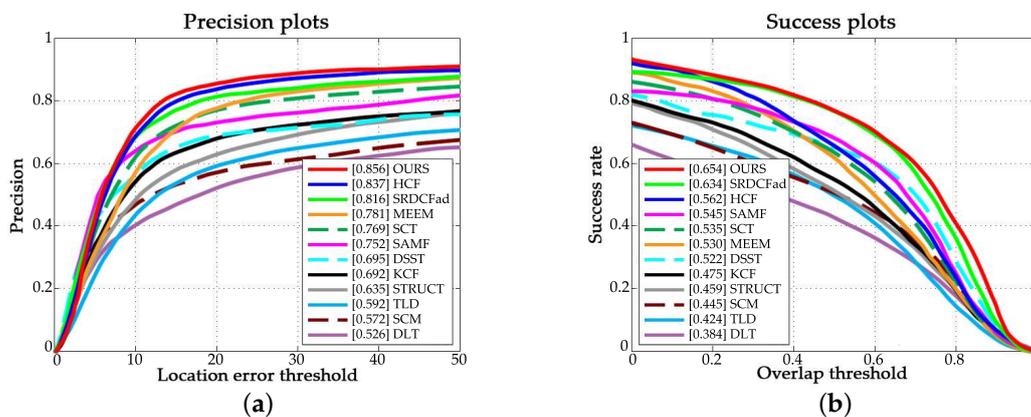
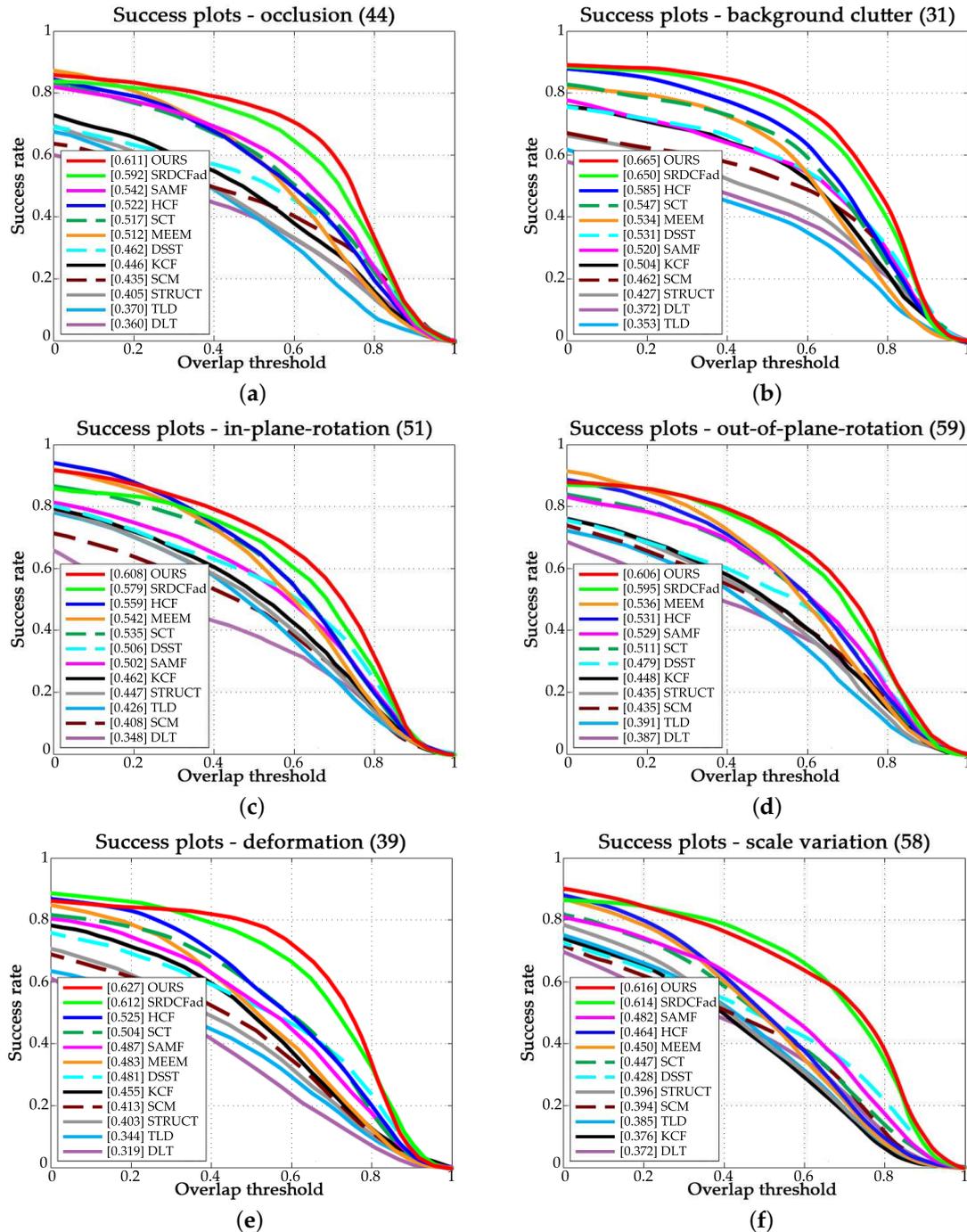


Figure 4. Precision and success plots over the OTB-2015 dataset. (a) Precision plot; (b) Success plots.

Attribute-based experimental results are shown in Figure 5. From Figure 5, we have the following observations. Firstly, our approach handled occlusion efficiently, which can be explained by the proposed adaptive model updating method. This method also helped to improve the performance in sequences with attribute of background cluster. Secondly, our approach performed well in the sequences with attributes of rotation and deformation, as the higher layers of CNN retain rich semantic information of the target object. Thirdly, our approach performs favorably against other approaches in sequences with the attribute of scale variation due to the rich texture information encoded in lower layers.



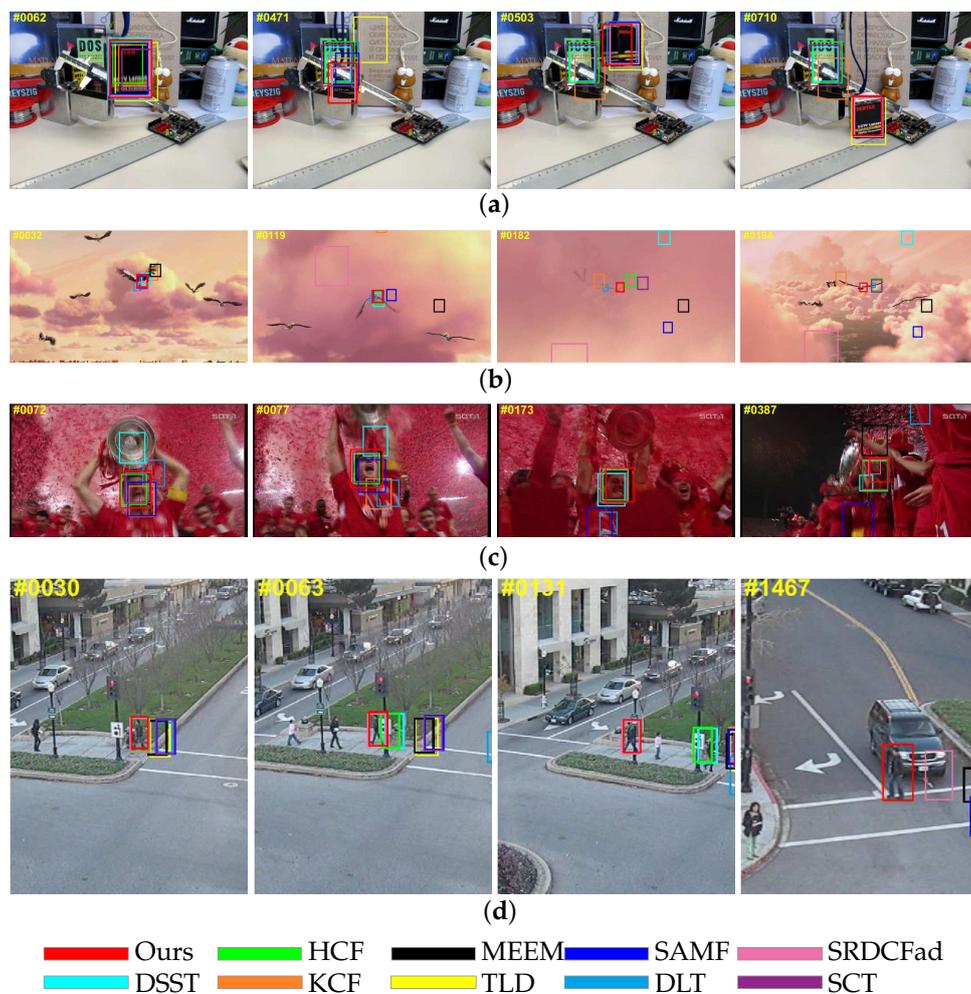
**Figure 5.** Success plots over six tracking challenges of (a) occlusion; (b) background clutter; (c) in-plane rotation; (d) out-of-plane; (e) deformation; and (f) motion blur.

## 4.2. Qualitative Evaluation

To better analyze the effectiveness and robustness of the proposed tracker, this section is divided into two subsections to conduct a qualitative analysis.

### 4.2.1. Performance against Background Information Variation

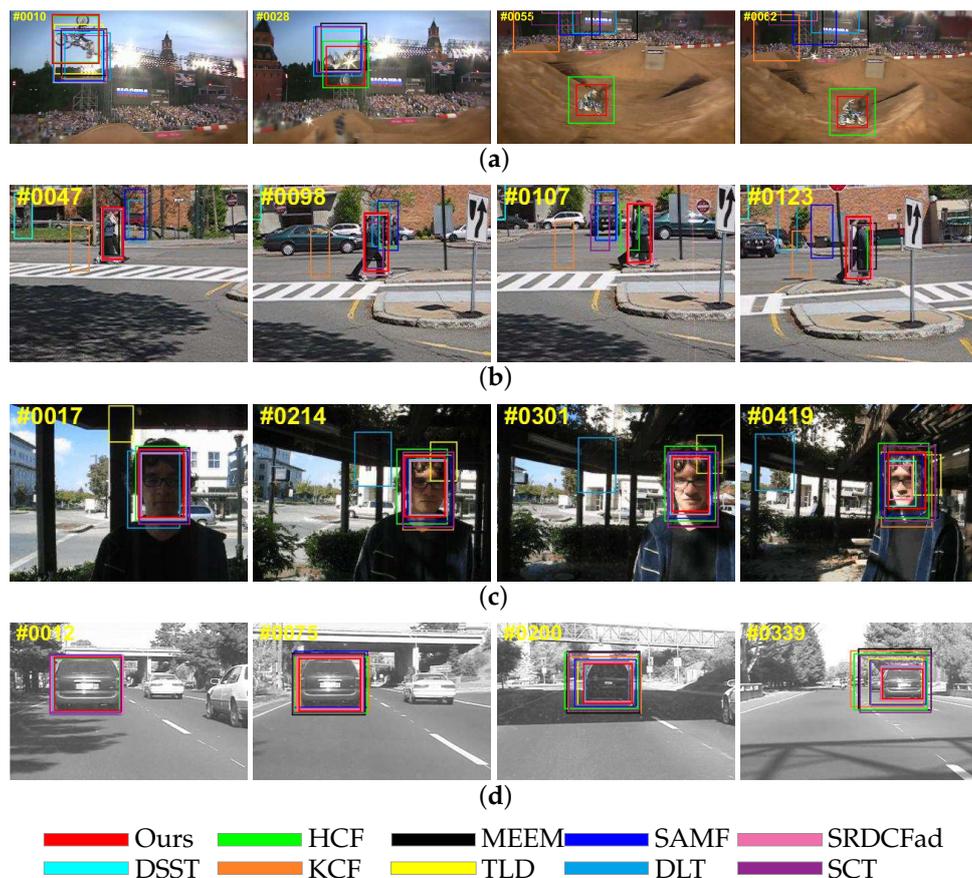
In this section, we focus on the trackers' performance against background information variation, including occlusion and background cluster. Figure 6 shows the situation in which the target undergoes severe occlusion and background cluster. In the sequence box and Bird1, the occlusion takes up to 35 frames and 50 frames, respectively. In the sequence Human3, the target is occluded by two different objects. In the sequence Soccer, spatial context information of the target changes dramatically and the target is nearly fully occluded. Due to the proposed updating method, the discounting factor decreases to near zero when the target undergoes occlusion, which prevents the correlation filter from learning occlusion information and losing the ability to discriminate the target. It should be noted that in the sequence Human3, only our tracker and SRDCFad succeeded in tracking the target at the first 1400 frames, which means that an adaptive model updating method is significant when the occlusion situation is complex. Besides, semantic information encoded in higher layers ensures that the tracker is not sensitive to background cluster.



**Figure 6.** Tracking results on sequences with attributes of occlusion and background cluster. From top to bottom, the name of the video is (a) Box; (b) Bird1; (c) Soccer; (d) Human3.

#### 4.2.2. Performance against Target Appearance Variation

In this section, we will discuss the trackers' performance against the variation of target appearance. In Figure 7a, the near-270-degree in-plane rotation of motorcyclists in the MotorRolling sequence is a big challenge for visual tracking algorithms. Since the high layers of CNN retain rich semantics information, our tracker performed well in this sequence. Although HCF and DLT also use convolutional features, these trackers cannot fully exploit fine-grained information, so HCF failed in scale estimation and DLT lost target totally. Similar results also appeared in the sequence Couple with the attribute of out-of-plane rotation (Figure 7b). In the sequences Trellis (Figure 7c) and Car4 (Figure 7d), the targets undergo scale variation and illumination variation at the same time, making it difficult to determine the precise boundary of the target. Since the fine-grained information in the lower layers is used properly, our tracker provided accurate scale estimation in these two sequences.



**Figure 7.** Tracking results on sequences with attributes of occlusion and background cluster. From top to bottom, the name of the video is (a) MotorRolling; (b) Couple; (c) Trellis; (d) Car4.

#### 4.3. Demonstrations

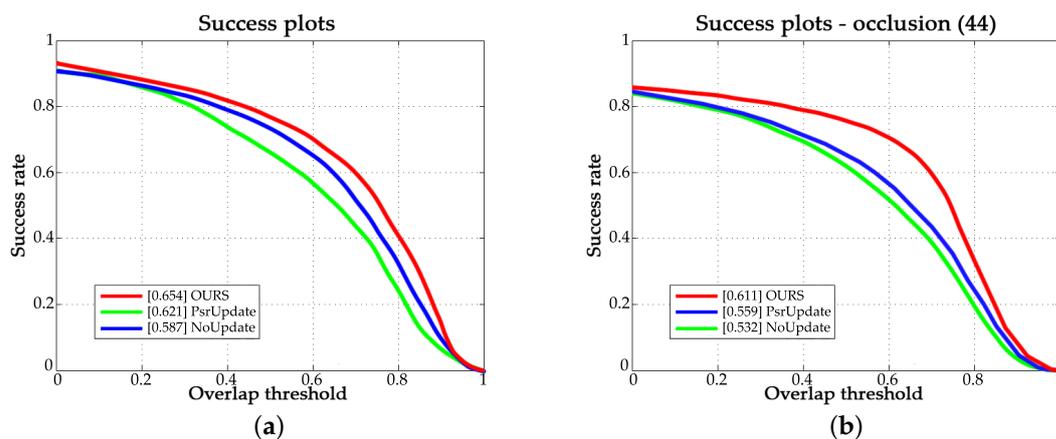
To evaluate the effect of updating method and scale estimation, we conducted additional comparison experiments on the OTB-2015 dataset.

##### 4.3.1. Evaluation of the Updating Method

We compared our method with the updating method using peak to sidelobe ratio (PSR) and updating method using fixed learning rate. The calculation of PSR is described in [13]. Let  $PSR_{mean}$  denote the average value of the historical data of the PSR. The application of PSR can be expressed by Equation (19):

$$\eta = \begin{cases} 0, & PSR < PSR_{mean} - 2 \\ \eta, & PSR \geq PSR_{mean} - 2 \end{cases}. \quad (19)$$

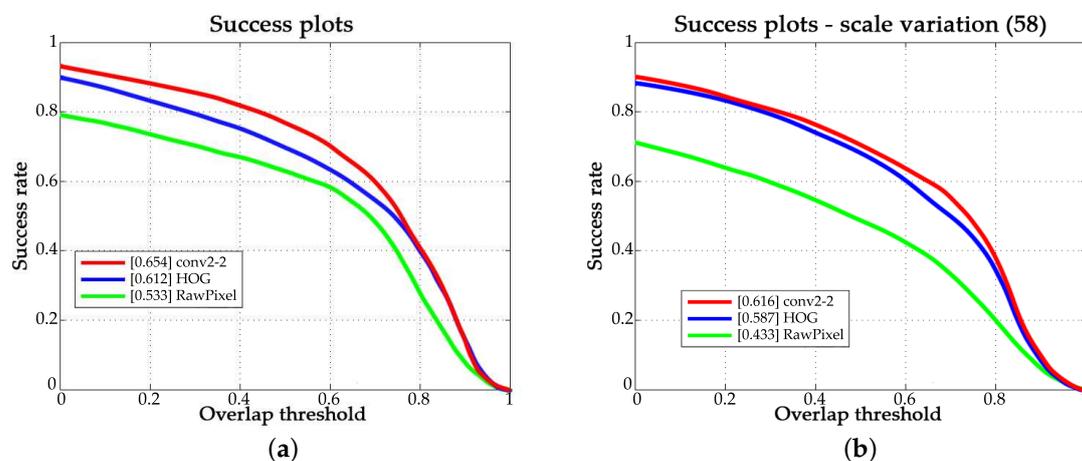
Experimental results are shown in Figure 8. ACMD is our proposed method. PsrUpdate is the same as ACMD, except it uses PSR as an update criterion. No update indicates that it uses a fixed learning rate update method. As shown in Figure 8a, ACMD led to 6.7% performance improvements in terms of success rate on OTB-2015 dataset. Besides, from Figure 8 we can find that the improvement of using PSR is limited. Moreover, as we can easily see in Figure 8b, the ACMD had a greater advantage over sequences with occlusion attribute, which illustrates the effectiveness of the proposed updating method in handling occlusion issues.



**Figure 8.** Tracking performance of different updating methods. (a) Success plots on OTB-2015; (b) Success plots over sequences with occlusion attribute.

#### 4.3.2. Evaluation of Scale Estimation

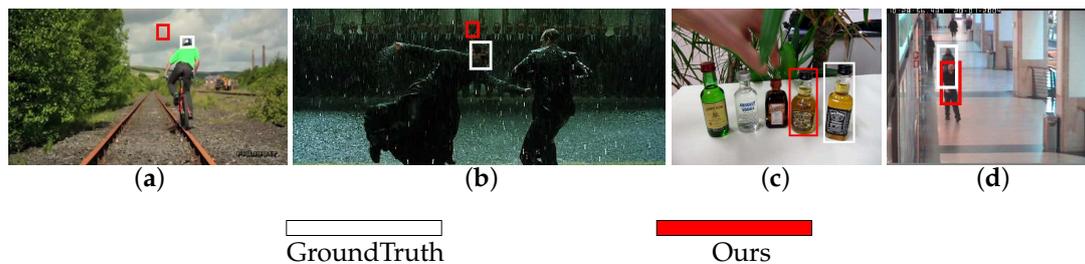
In order to evaluate the performance of convolutional features for scale estimation, we used the HOG feature and the raw pixel feature as comparisons. The results are shown in Figure 9. The legend in Figure 9 annotates the feature used in the corresponding curve. From Figure 9, we can see that scale estimation using convolution feature works best in scale estimation. In the success rate evaluation over all 100 sequences and 58 sequences with attribute of scale variation, convolutional feature-based scale estimation led to 23.9% and 19.7% performance improvement, respectively.



**Figure 9.** Scale estimation performance using different features. (a) Success plots on OTB-2015; (b) Success plots over sequences with the attribute of scale variation.

#### 4.4. Failure Cases

We show a few failure cases in Figure 10. For the Biker and Matrix sequences, when the target object undergoes fast motion, the proposed tracker fails to follow targets due to the boundary effect introduced in the correlation model. For the Liquor and Walking2 sequences, the proposed method fails to track the target as the target is occluded by a similar object and the the discounting factor cannot decrease when occlusion occurs. Correlation filters with less boundary effect and strategies for handling similar object interference will be considered in our future work.



**Figure 10.** Failure cases on the following sequences: (a) Biker; (b) Matrix; (c) Liquor; (d) Walking2.

## 5. Conclusions

In this paper, we propose a visual tracking framework which synthesizes features from multiple layers in a CNN and makes full use of the VGG network. The proposed tracker can make precise position estimation in many challenging videos. The novel model updating method of the tracker improves the tracking performance in occlusion scenarios. Moreover, the use of convolutional features ensures the accuracy of scale estimation. Numerous experimental results demonstrate that the proposed tracker outperforms the state-of-the-art trackers in both precision and success rate.

**Acknowledgments:** This work was supported by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802, and the General Program of National Nature Science Foundation of China under Grants 61371132 and 61471043.

**Author Contributions:** Yuankun Li and Tingfa Xu designed the tracking algorithm and the corresponding experiments. Yuankun Li and Guokai Shi accomplished the MATLAB codes of the experiments. Yuankun Li, Honggao Deng and Jie Guo analyzed the data. Yuankun Li wrote the paper.

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

## References

1. Babenko, B.; Yang, M.H.; Belongie, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632.
2. Danelljan, M.; Khan, F.S.; Felsberg, M.; Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1090–1097.
3. Nebel, G.; Pflugfelder, R. Clustering of static-adaptive correspondences for deformable object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2784–2791.
4. 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.
5. 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, Washington, DC, USA, 7–13 December 2015; pp. 3074–3082.
6. Danelljan, M.; Häger, G.; Khan, F.S.; 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.

7. Wang, N.; Yeung, D.Y. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems*; NIPS: La Jolla, CA, USA, 2013; pp. 809–817.
8. Wang, N.; Li, S.; Gupta, A.; Yeung, D. Transferring Rich Feature Hierarchies for Robust Visual Tracking. *arXiv* **2015**, arXiv: 1501.04587.
9. Li, H.; Li, Y.; Porikli, F. DeepTrack: Learning Discriminative Feature Representations by Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; p. 3.
10. Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.H. Robust Visual Tracking via Convolutional Networks Without Training. *IEEE Trans. Image Proc.* **2016**, *25*, 1779–1792.
11. 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.
12. 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.
13. 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 Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
14. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-detection with Kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715.
15. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. 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.
16. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 4310–4318.
17. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
18. Adam, A.; Rivlin, E.; Shimshoni, I. Robust Fragments-based Tracking using the Integral Histogram. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–23 June 2006; pp. 798–805.
19. Zhang, T.; Jia, K.; Xu, C.; Ma, Y.; Ahuja, N. Partial Occlusion Handling for Visual Tracking via Robust Part Matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1258–1265.
20. Lowe, D.G.; Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
21. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust Invariant Scalable Keypoints. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
22. Grabner, M.; Grabner, H.; Bischof, H. Learning Features for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
23. Hare, S.; Saffari, A.; Torr, P.H.S. Efficient Online Structured Output Learning for Keypoint-Based Object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1894–1901.
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv: 1409.1556.
25. Liu, L.; Shen, C.; van den Hengel, A. The Treasure Beneath Convolutional Layers: Cross-Convolutional-Layer pooling for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 17–22 June 2015; pp. 4749–4757.
26. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

27. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 430–443.
28. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
29. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848.
30. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 1430–1438.
31. Choi, J.; Chang, H.J.; Jeong, J.; Demiris, Y.; Choi, J.Y. Visual Tracking Using Attention-Modulated Disintegration and Integration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 4321–4330.
32. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 188–203.
33. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the European Conference on Computer Vision Workshop, Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
34. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H.S. Struck: Structured Output Tracking with Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109.
35. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422.
36. Zhong, W.; Lu, H.; Yang, M.H. Robust Object Tracking via Sparse Collaborative Appearance Model. *IEEE Trans. Image Proc.* **2014**, *23*, 2356–2368.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).