

Article

# Partial Block Scheme and Adaptive Update Model for Kernelized Correlation Filters-Based Object Tracking

Soowoong Jeong and Joonki Paik \* 

Department of Image, Chung-Ang University, Seoul 06974, Korea; imgrecog@gmail.com

\* Correspondence: paikj@cau.ac.kr; Tel.: +82-10-7123-6846

Received: 30 May 2018; Accepted: 7 August 2018; Published: 10 August 2018



**Abstract:** In visual object tracking, the dynamic environment is a challenging issue. Partial occlusion and scale variation are typical challenging problems. We present a correlation-based object tracking based on the discriminative model. To attenuate the influence by partial occlusion, partial sub-blocks are constructed from the original block, and each of them operates independently. The scale space is employed to deal with scale variation using a feature pyramid. We also present an adaptive update model with a weighting function to calculate the frame-adaptive learning rate. Theoretical analysis and experimental results demonstrate that the proposed method can robustly track drastic deformed objects. The sparse update reduces the computational cost for real-time tracking. Although the partial block scheme generation increases the computational cost, we present a novel sparse update approach to reduce the computational cost drastically for real-time tracking. The experiments were performed on a variety of sequences, and the proposed method exhibited better performance compared with the state-of-the-art trackers.

**Keywords:** computer vision; object tracking; correlation filter; partial block; scale space; adaptive learning; discriminative model; partial occlusion; scale variation

## 1. Introduction

Tracking the position of objects of interest from a sequence of video frames is a fundamental problem in computer vision research. Object tracking is an integral part of computer vision and is applied in various fields including robotics, surveillance system, motion analysis, autonomous cars, unmanned aerial vehicles (UAVs) and human computer interaction (HCI). However, the research on object tracking is still recognized as a difficult problem since the object tracking environment contains various challenging factors, such as illumination variation, scale variation, occlusion, deformation, motion blur, fast motion and rotation. These factors significantly degrade the performance of object tracking. For that reason, minimizing the influence of environmental changes in the development of robust trackers is an important issue. There are many tracking algorithms [1–37] to deal with the variety of environmental changes. The state-of-the-art tracking algorithms have tried to solve the problem by analyzing the cause of environmental changes using various classification approaches.

In this paper, we present a novel correlation filter-based object tracking algorithm that focuses on solving scale variation and partial occlusion problems. The proposed algorithm is based on a discriminative model tracker with a correlation filter. The kernelized correlation filter (KCF) tracker [21] has demonstrated outstanding performance for object tracking by drastically reducing computational cost using an efficient search based on the diagonalization property of a circular matrix and a dual correlation filter (DCF). However, the KCF tracker is sensitive to environment changes because it still does not consider partial occlusion and scale variation, which make the performance of the tracker poor.

Most tracking-by-detection algorithms consider only object translation [22], but the proposed algorithm deals with scaling and partial occlusion, as well as object translation. Partial occlusion is a significant problem that degrades the performance of object detection. We propose a robust KCF-based tracker to overcome the partial occlusion problem using a partial block scheme. The partial block scheme facilitates stable object tracking, even if the object is partially occluded. A robust tracker also needs a strategy for scale estimation to deal with changes of the object size. The scale space [22] creates an image pyramid and determines the most appropriate size of an object block. We also propose an adaptive update model using a weighting function by improving the general update model used in the original version of the kernelized correlation filter [21]. The proposed adaptive update model calculates the learning rate with a modified sigmoid function as the weighting function, and then, the optimal learning rate is calculated for each frame.

In summary, the proposed method is developed to deal with partial occlusion, scaling, illumination variation and deformation. Experimental results demonstrated that the proposed method exhibits a better performance than existing state-of-the-art algorithms for various test videos including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion and rotation. Figure 1 compares the performances of the proposed method and state-of-the-art trackers. More specifically, Tiger2 including partial occlusion substantiates that the proposed partial block scheme successfully solves the partial occlusion problem. Freeman3 and Shaking substantiate that the performance of the proposed method is good enough for the scale variation problem. On the other hand, existing trackers do not properly respond to occlusion and scale variation problems.



**Figure 1.** Comparison of the proposed method with the state-of-the-art trackers kernelized correlation filter (KCF) [21], discriminative scale space tracker (DSST) [22], fragment-based tracker (FRAG) [15], locality sensitive histograms tracker (LSHT) [37], multiple instance learning (MIL) [6], structured output tracking with kernels (STRUCK) [5] and tracking-learning-detection (TLD) [9]. The sequences include Tiger2, Freeman3 and Shaking from OTB-100.

The composition of this paper is as follows: Section 2 describes the technical background of tracking with related works. Section 3 presents the proposed method with the object translation estimation using the partial block scheme, the object scaling estimation using scale space and the adaptive update model. After summarizing experimental results in Section 4, Section 5 concludes the paper.

## 2. Related Works

The existing trackers [1–37] have different approaches and can be classified according to the model. In recent years, the trackers have been divided into discriminative model-based [1–10,19–22] and generative model-based [11–18], and the proposed method belong to the discriminative model. Furthermore, some discriminative model-based trackers can be classified as correlation-based trackers [19–22].

The generation model-based trackers use a method of modeling the appearance of an object of interest and have various models for representing the object. Among generation model-based trackers, incremental visual tracking (IVT) [11] uses a PCA and applies an adaptive appearance update model to withstand lighting changes and variations. However, IVT is very sensitive to partial occlusion where the object is partially obscured by other objects. The occlusion problem was improved by applying the probability continuous outlier model (PCOM) [12] based on IVT. The visual tracking by decomposition (VTD) tracker [13] extended particle filter tracking (PFT), and the L1-minimization tracker [14] employed sparse representation. Furthermore, the fragment-based tracker (FRAG) [15] employed the local patch to ensure robustness for solving the partial occlusion problem, and the circulant sparse tracker (CST) [16] employed a combination of circularity and scarcity expressions. In addition, multi-task sparse learning tracker (MTT) [17] and the low-rank sparse trackers [18] belong to the generation model-based trackers.

The discriminative model employs the method that classifies objects and background, and it learn the model directly. The ensemble trackers [1] proposed to combine multiple weak classifiers to form an ensemble structure. The online Ada boosting (OAB) [2] employed identifiable feature selection and online boosting, and the online random forest (ORF) [3] learned and classified random forests online. STRUCK [5], using the kernel [6], online support vector machine (SVM) [4] and the multiple instance learning (MIL) tracker with HAAR features also belong to the discriminative model. The weighted multiple instance learning (WMIL) [7] improved the weighting of positive samples in the MIL and reflected the weighting of samples when learning the classifier. The correlation filter-based tracker also belongs to the discriminative model. The minimizing the output sum of squared error (MOSSE) tracker [19] proposed an adaptive correlation filter, and circulant structure of circulant structure kernel (CSK) [20] used the dense sampling with the theory of circulant matrices and fast Fourier transform (FFT). Furthermore, the kernelized correlation filter (KCF) tracker [21] applied linear and kernel ridge regression with histogram of oriented gradients (HOG) features for high-speed tracking. However, the KCF tracker can only be used for object translation estimation. The scale estimation problem was solved by the discriminative scale space tracker (DSST) [22], which estimates the translation and scale, independently.

### *Discriminative Correlation Filter*

In recent research, discriminative classifiers were the core component of modern trackers, and the discriminative model distinguished the object from the surrounding environment [21] to effectively track the object of interest. To distinguish between the object and surrounding environment, discriminative model-based trackers [1–10,19–22] learned about the positive samples and the negative samples. The discriminative model-based trackers were considered to be more significant with respect to negative samples, and negative samples did not cover the object, completely. This means that the positive sample was located closer to the location of the object, and the positive sample contained enough information to represent the object. Generally, a large number of negative samples increases the

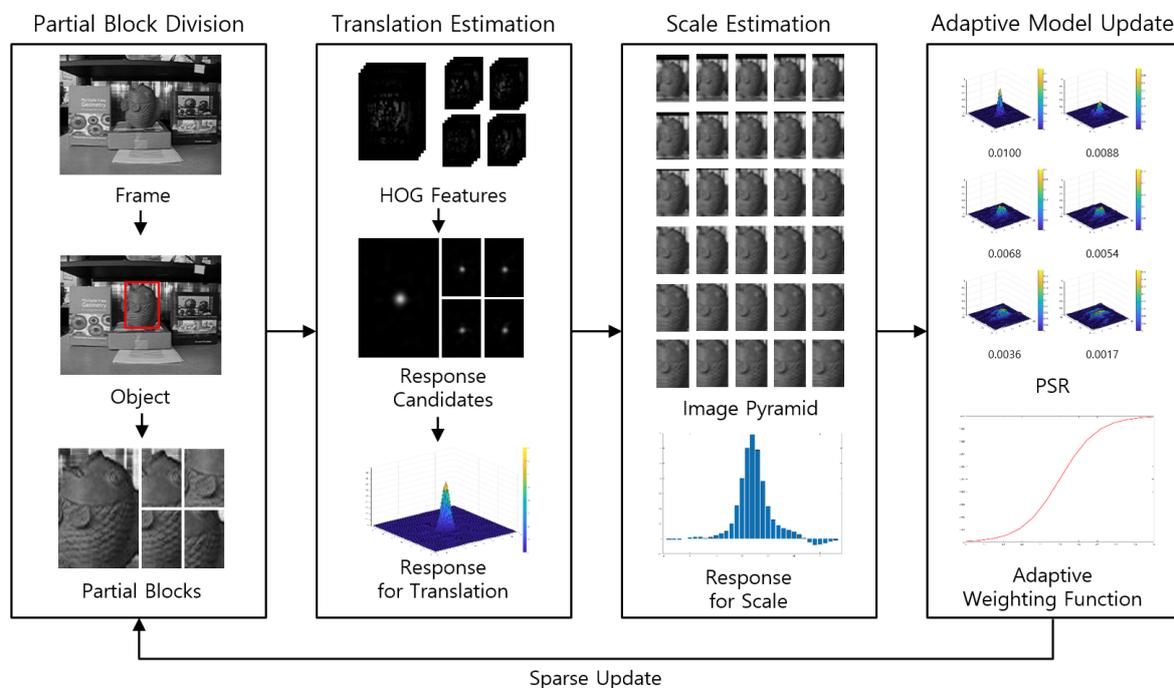
computational cost. Most trackers [3,4,6,8,9,32] employed the random sampling methods to avoid high computational cost. However, correlation filter-based trackers [20–22] efficiently tracked objects using the circulant structure [20] and FFT to incorporate all samples without iterating them. In addition, the dual correlation filter (DCF) was proposed in the literature [21]. The DCF performs linear multi-channel filtering for a similar performance of a nonlinear kernel with very low complexity. Thus, CSK [20], KCF [21] and DSST [22] used dense sampling with all samples; nevertheless, they can perform object tracking in real time because the computational cost is not high. Real-time processing is one of the significant components in object tracking for various vision applications.

### 3. Proposed Method

The partial occlusion and scale variation in object tracking comprise a crucial problem. Many researchers have tried to solve this problem, but it is still known to be a difficult problem. In addition, object tracking in real-time is also significant because its target is videos. The real-time object tracking that these problems have solved can be applied to many vision applications. Therefore, our goal is to develop real-time object tracking with the consideration of partial occlusion and scale variation.

The proposed work is based on KCF, which consists of (i) the detection part for describing objects and (ii) the training part for learning. The updated model is used in the detection part of the next frame, and the entire process repeats to track objects continuously.

In this paper, we propose object tracking using the partial block scheme and the adaptive update model. The partial block scheme is proposed to solve the partial occlusion problem, and the adaptive update model employs the weighted learning rate. The weighting is calculated from the reliability of the response of each block with a sigmoid function. If the reliability of the response is high, we use a higher learning rate. On the other hand, if the reliability of the response is low, we use a lower learning rate. Furthermore, a sparse update is performed to reduce the increased computation due to multiple partial blocks. Figure 2 shows the block diagram of the proposed method.



**Figure 2.** The block diagram of the proposed method. We separate the partial blocks from the object in the frame and perform the translation estimation and scale estimation. Finally, we perform the model update with the weighting function or skip the update. PSR, peak-to-sidelobe ratio.

The proposed methods can be divided into four steps as follows:

1. Partial block separation: separating the partial blocks from the whole block of an object. Partial blocks can be adjusted in size and position according to the parameter.
2. Translation estimation: calculating the responses using a kernelized correlation filter of all blocks and then selecting the translation response map.
3. Scale estimation: estimating the object scale with the scale space and calculating the scale factor.
4. Adaptive model update: model updating with the adaptive learning rate considering the reliability of responses.

### 3.1. Partial Block Scheme

We propose the partial block scheme to address environmental changes such as partial occlusion, partial illumination variation and partial blurring. Partial blocks are computed for each frame from the whole block and divided into four parts. We can generate the partial blocks using the whole block and Equation (1).

$$\begin{aligned} P^m &= W^m / d \\ P^n &= W^n / d \end{aligned} \tag{1}$$

where  $W^m$  and  $W^n$  are the height and width of the whole block,  $P^m$  and  $P^n$  are the height and width of partial blocks and the sizes of partial blocks are identical.  $d$  is a factor that adjusts the partial blocks' size. The centers of partial blocks are obtain with:

$$B_k(x_c, y_c) = \begin{cases} W(x_c, y_c), & k = 0 \\ W(x_c, y_c - \omega P^m), & k = 1 \\ W(x_c, y_c + \omega P^m), & k = 2 \\ W(x_c - \omega P^n, y_c), & k = 3 \\ W(x_c + \omega P^n, y_c), & k = 4 \end{cases} \tag{2}$$

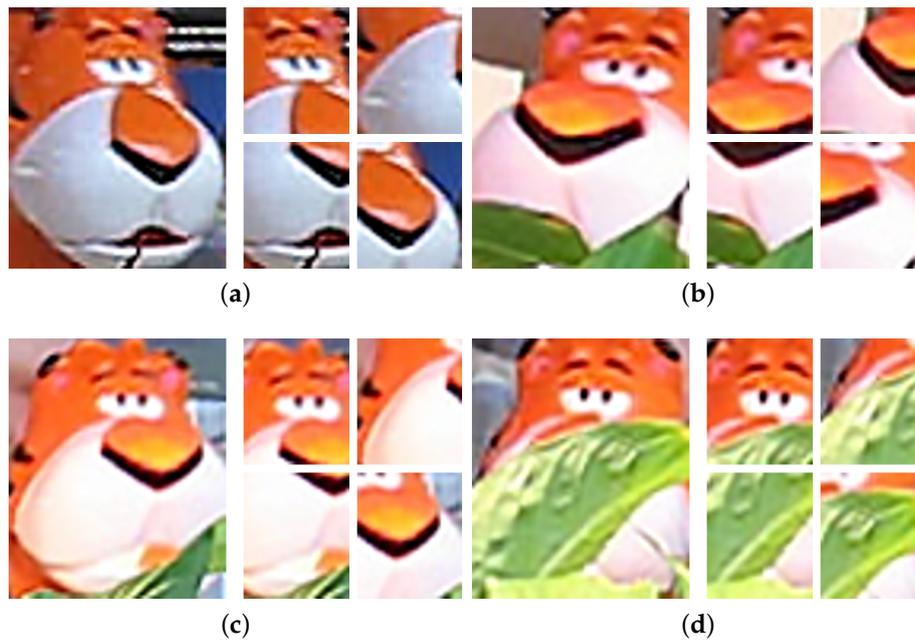
In Equation (2),  $W(x_c, y_c)$  is the center position of the whole block, and  $B_k(x_c, y_c)$  is the center position of all blocks, which include the whole block and partial blocks.  $k$  means the index of blocks, and  $\omega$  means the factor that adjust the location for partial blocks. The indices of partial blocks are 0–4, and  $B_0$  means the whole block.  $B_1, B_2, B_3$  and  $B_4$  mean partial blocks, respectively. As shown Figure 3, the positions of partial blocks depend on the position of the whole block, and this can adjust the parameter  $\omega$ . Furthermore, the sizes of the partial blocks are set to an identical size for the convenience of calculation by parameter  $d$ . We proposed a partial block scheme to deal with the partial occlusion problem. Partial occlusion can occur in all blocks including the whole block. However, the proposed method is designed to track any block without partial occlusion.

The whole block is small or unsuitable parameter  $d$  can produce too small partial blocks. The small size of partial blocks among the generated partial blocks can disturb object tracking. Thus, we employ excluding very small blocks using Equations (3) and (4).

$$B_{k \in \{1,2\}}^w = \begin{cases} 0, & P^m < \tau \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

$$B_{k \in \{3,4\}}^w = \begin{cases} 0, & P^n < \tau \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

where  $\tau$  is the threshold for the decision whether to exclude partial blocks.  $B_k^w$  means the weighting for partial blocks. If  $P^m$  is smaller than  $\tau$ ,  $B_1^w$  and  $B_2^w$  are excluded blocks. Furthermore, if  $P^n$  is smaller than  $\tau$ ,  $B_3^w$  and  $B_4^w$  are excluded blocks. All partial blocks are not large enough; we can only use whole blocks, and  $B_0^w$  is always 1.



**Figure 3.** The whole block and its partial blocks from Tiger2 in OTB-100. (a) 1st frame without occlusion; (b) 95th frame with slight occlusion; (c) 115th frame with slight occlusion; (d) 256th frame with large occlusion.

### 3.2. Translation Estimation

The kernelized correlation filter (KCF) tracker [21] is a representative tracking algorithm based on correlation filters; it is superior in terms of performance and speed. The correlation filter tracker aims to calculate a filter  $h$  that minimizes the square error of sample data and regression data. The KCF tracker calculates filter  $h$  between sample data  $x_i$  and regression data  $y_i$  with:

$$\min_h \sum_{i=1}^n (h^T x_i - y_i)^2 + \lambda \|h\|^2 \tag{5}$$

where  $\lambda$  is the regularization parameter. In the KCF tracker, we employ the kernel trick [38] for the non-linear regression function. Thus, non-linear filters could be as fast as linear correlation filters. The kernelized version of ridge regression is defined [21] as:

$$\alpha = (K + \lambda I)^{-1} y. \tag{6}$$

where  $K$  is the kernel matrix,  $I$  is the identity matrix and  $\alpha$  is the represented vector [20,21] of filter  $h$ . The  $n \times n$  kernel matrix can be expressed in the circulant matrix [21] as follows.

$$C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \tag{7}$$

The circular structure can express the same signal  $x_n$  according to  $n$  shift due to periodic characteristics. In Equation (7), the first row  $\{x_1, \dots, x_n\}$  is the base samples, and cyclically-shifted rows are virtual samples.

The kernel matrix can be diagonalized by discrete Fourier transform (DFT), and the kernel ridge regression solution is defined [21] by:

$$\alpha = F^{-1} \left( \frac{F(y)}{F(\kappa) + \lambda} \right). \tag{8}$$

Equation (8) is a closed-form solution, which is very efficient; it uses only fast Fourier transform (FFT) and element-wise operation [20]. The following equation [21] calculates  $\kappa$ ,

$$\kappa^{xx'} = \exp \left( -\frac{1}{\sigma^2} (\|x\|^2 + \|x'\|^2) - 2F^{-1} \left( \sum_c F(x) \odot F^*(x') \right) \right) \tag{9}$$

where  $\odot$  indicates the element-wise product and  $\kappa^{xx'}$  is the kernel correlation of  $x$ ; it can be computed quickly with FFT [20].  $F$  means the fast Fourier transform, and  $F^{-1}$  means the inverse transform.

In this paper, we perform the kernel ridge regression solution for all blocks using the equation below.

$$R = F^{-1}(F(\hat{\alpha}) \cdot F(\kappa^{xz})) \tag{10}$$

where  $\hat{\alpha}$  means the update model of  $\alpha$ ,  $R$  means the response map and  $R_0$  is for the whole block.  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$  are the response maps for partial blocks, respectively. Furthermore, we perform the weighting for each response map and pick the suitable response map by:

$$k^* = \arg \max_{k \in \{0,1,2,3,4\}} (\max(R_k \cdot B_k^w)). \tag{11}$$

The index for all blocks is  $k$ ;  $k^*$  means the index of the picked response that includes the highest value of all response maps. As we mentioned before,  $B_k^w$  is the weighting for blocks. The position of the highest value in the picked response map means the translated position of the object. Then, we can calculate the center position for the picked block with  $\Delta x$ ,  $\Delta y$  by:

$$\hat{B}_k(x_c, y_c) = B_{k^*}(x_c + \Delta x, y_c + \Delta y). \tag{12}$$

We recalculate the central position for the new whole block using  $\hat{B}_k(x_c, y_c)$ .

$$\hat{W}(x_c, y_c) = \begin{cases} \hat{B}_0(x_c, y_c), & k^* = 0 \\ \hat{B}_1(x_c, y_c + \omega(\eta^* P^m)), & k^* = 1 \\ \hat{B}_2(x_c, y_c - \omega(\eta^* P^m)), & k^* = 2 \\ \hat{B}_3(x_c + \omega(\eta^* P^n), y_c), & k^* = 3 \\ \hat{B}_4(x_c - \omega(\eta^* P^n), y_c), & k^* = 4 \end{cases} \tag{13}$$

where  $\hat{W}(x_c, y_c)$  is the updated center position of the whole block and  $\eta^*$  means the scale factor. Then, in the next frame, we can obtain new partial blocks using Equation (2).

### 3.3. Scale Estimation

The translation estimation tracks the horizontal and vertical movements of objects. Thus, tracking only the translation of the object has limited performance object tracking. The DSST [22] proposed scale space for accurate scale estimation. Scale space expresses the data in 3 dimensions; the size is  $P^m \times P^n \times D$ . Here,  $P^m$ ,  $P^n$  and  $D$  are the height, width and dimension, respectively. We compose the image pyramid for scale estimation with  $\hat{W}(x_c, y_c)$ . The equation for the composition of the image pyramid consisting of various sizes is as follows.

$$\eta_l = s^c, c \in \left\{ \left\lfloor \frac{D}{2} \right\rfloor - 1, \dots, \left\lfloor \frac{D}{2} \right\rfloor - D \right\} \tag{14}$$

The scale factors  $\eta_l$  contains large values to small values to compose the image pyramid.  $s$  is the factor for the scale step. The dimension of features is defined by  $l \in \{1, \dots, D\}$ . After the image pyramid has been composed, we can calculate the scale response using the equation below [22].

$$\beta = F^{-1} \left( \frac{\sum_{l=1}^D F(A_l) \cdot F(Z_l)}{F(S) + \lambda'} \right) \tag{15}$$

where  $Z_l$  means the object,  $A_l$  is the desired output and  $S$  is the kernel correlation result. The scale response  $\beta$  has  $D$ -th values, and the index of highest value means scale factor  $\eta^*$ .

$$\eta^* = \arg \max_{l \in \{1, \dots, D\}} (\max(\beta_l)) \tag{16}$$

The picked scale factor  $\eta^*$  is multiplied by the size and center position of all the blocks.

### 3.4. Adaptive Update Model

In the tracking process, the attributes of objects change constantly. Furthermore, most objects maintain continuity with the previous frame. We propose that the adaptive learning rate is applied using the reliability of the response map. The peak-to-sidelobe ratio (PSR) [19] value of the response regards the reliability of the response map. It reflects the relationship between the main lobe and the surrounding side lobe by:

$$\rho_k = \psi \left( \frac{R_k^p - R_k^\mu}{R_k^\sigma} \right) \tag{17}$$

where  $R_k^\mu$ ,  $R_k^\sigma$  and  $R_k^p$  are the mean, standard deviation and peak value of the response.  $\psi$  is a regularization parameter. The weighting function calculates the adaptive learning rate  $\hat{\rho}_k$ .

$$\hat{\rho}_k = \left( \frac{\rho_m}{1 + e^{-v_1 \cdot (\rho_k - v_2)}} \right)^\gamma \tag{18}$$

where  $v_1$ ,  $v_2$  and  $\gamma$  are the parameters of the weighting function and  $\rho_m$  controls the maximum learning rate.

Figure 4 shows the response maps for two different values of the adaptive learning rate. The reliability of the response map determines the adaptive learning rate value. Specifically, the higher the rate value, the more it influences the update model, and vice versa. The adaptive update model is defined by:

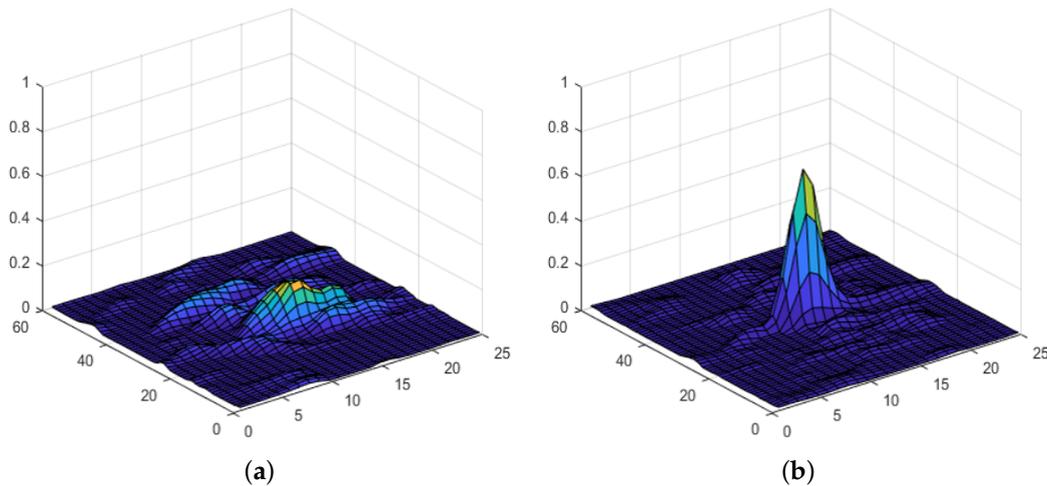
$$\hat{\alpha}_t = (1 - \hat{\rho})\hat{\alpha}_{t-1} + \hat{\rho} \cdot \alpha_t. \tag{19}$$

The number of  $\hat{\rho}$  depends on the number of blocks.  $\hat{\alpha}_t$  means the update model for the current frame, and  $\hat{\alpha}_{t-1}$  means the update model for the previous frame.  $\alpha_t$  is the calculated kernel regression solution in the current frame. The adaptive update model is performed to learn the translation estimation and scale estimation, identically.

For real-time tracking, we assume that the peak position of the response is the center, and we do not need to calculate the update model for the next frame. Then, we employ the sparse update for real-time object tracking. The sparse update is given by:

$$\delta = \begin{cases} true, & |\Delta x + \Delta y| \leq \hat{\tau} \\ false, & otherwise \end{cases} \quad (20)$$

where  $\hat{\tau}$  is the parameter for the sparse update; when  $\delta$  is true, we can skip the model update for efficiency.



**Figure 4.** The adaptive learning rate according to response. (a) response map with learning rate value of 0.0055; (b) response map with learning rate value of 0.0289.

## 4. Experiments

We evaluated the proposed tracker with state-of-the-art trackers such as KCF [21], DSST [22], FRAG [15], LSHT [37], MIL [6], STRUCK [5] and TLD [9] for quantitative performance evaluation. The experiment was conducted with challenging sequences in the OTB-100 [39], and it included various attributes for natural sequences. Furthermore, we conducted experiments using one-pass evaluation (OPE). OPE is a general performance evaluation method used by the object tracking benchmark (OTB) [39].

### 4.1. Parameters and Experimental Setup

The factor  $d$  that adjusts the partial blocks size was set to two. The factor  $\omega$  that adjusts the location for partial blocks was set to 0.5. The excluded partial blocks decided by  $\tau$  were 15 pixels. The regularization parameter  $\lambda$  was 0.001. We used the scale step  $s = 1.02$  and the dimension of scale space  $D = 33$ . The regularization parameter for adaptive learning rate  $\psi$  was  $1/14$ .  $v_1$ ,  $v_2$  and  $\gamma$  were set to 10, 0.5 and 1.5.  $\rho_m$  was set to 0.03 for the maximum learning rate.  $\hat{\tau}$  was set to zero for the sparse update. In addition, we used histogram of oriented gradients (HOG) [40] as a feature to represent images. We conducted the experiments using MATLAB R2017b with an i7-2600 core 3.40-GHz CPU with 16 GB RAM.

#### 4.2. Quantitative Evaluation

We calculated the center location error (CLE) for quantitative performance evaluation; this means the euclidean distance between the center location of the ground truth and the estimated center location by the object tracker. The euclidean distance can be calculated as follows:

$$CLE = \frac{1}{N} \sum_{n=1}^N \sqrt{(x_b^n - x_g^n)^2 + (y_b^n - y_g^n)^2}. \quad (21)$$

where  $x_b^n$  and  $y_b^n$  are the estimated center location by the tracker and  $x_g^n$  and  $y_g^n$  are the center location from the ground truth.  $N$  means the total number of pixels in the bounding box. If the center positions are close, the tracker can obtain a lower value, which means the good performance of the tracker. The precision was defined as success within 20 pixels (threshold), otherwise precision was defined as a failure. Thus, precision depended on the CLE results and its threshold.

As another measurement, the success rate was defined as:

$$S = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}, \quad (22)$$

where  $r_t$  and  $r_a$  are bounding boxes for the ground truth and estimated results by the tracking algorithm.  $\cap$  and  $\cup$  indicate intersection and union. The function  $|\cdot|$  means the number of pixels in the bounding box. The higher the success rate, the more overlap between the estimated bounding box and the ground truth. The success rate varied according to the threshold, and the threshold used in the experiment was 0.5. We can draw a success rate graph considering all thresholds and define the under area of the graph as the area under the curve (AUC). Then, we conducted experiments for all sequences in OTB-100. Table 1 shows the average performance evaluation results of proposed method with the state-of-the-art trackers. The scores of the proposed method were highest for all measurements. The precision score was higher than KCF [21] and STRUCK [5], and the success rate and AUC were higher than DSST [22]. Experimental results demonstrated the effectiveness of the proposed method compared to existing trackers. The results showed that the proposed tracker was suitable for sequences including a variety of environments. The proposed tracker can be applied to a variety of vision applications such as robotics, surveillance systems, motion analysis, autonomous cars, unmanned aerial vehicles (UAVs) and human computer interaction (HCI), as mentioned above.

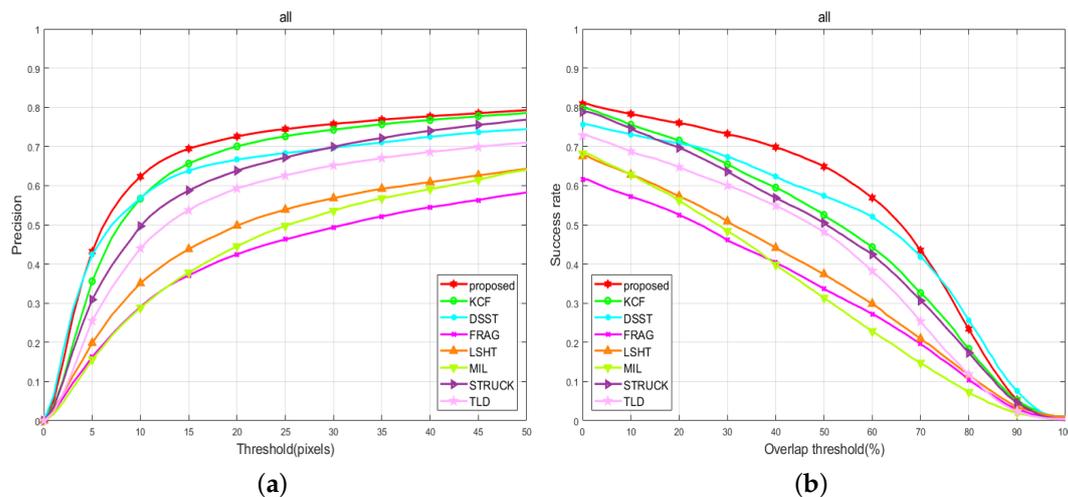
**Table 1.** The average performance evaluation results of trackers with the proposed method for 100 sequences in the OTB-100 database. The bold values mean the best performance. CLE, center location error.

Trackers	CLE	Precision	Success Rate	AUC
STRUCK [5]	47.09	0.6381	0.5046	0.4454
MIL [6]	71.96	0.4450	0.3132	0.3162
TLD [9]	60.14	0.5930	0.4819	0.4071
FRAG [15]	80.65	0.4245	0.3368	0.3182
LSHT [37]	68.24	0.4979	0.3742	0.3493
KCF [21]	44.88	0.7002	0.5252	0.4613
DSST [22]	56.47	0.6664	0.5738	0.4923
Proposed	<b>43.50</b>	<b>0.7253</b>	<b>0.6485</b>	<b>0.5280</b>

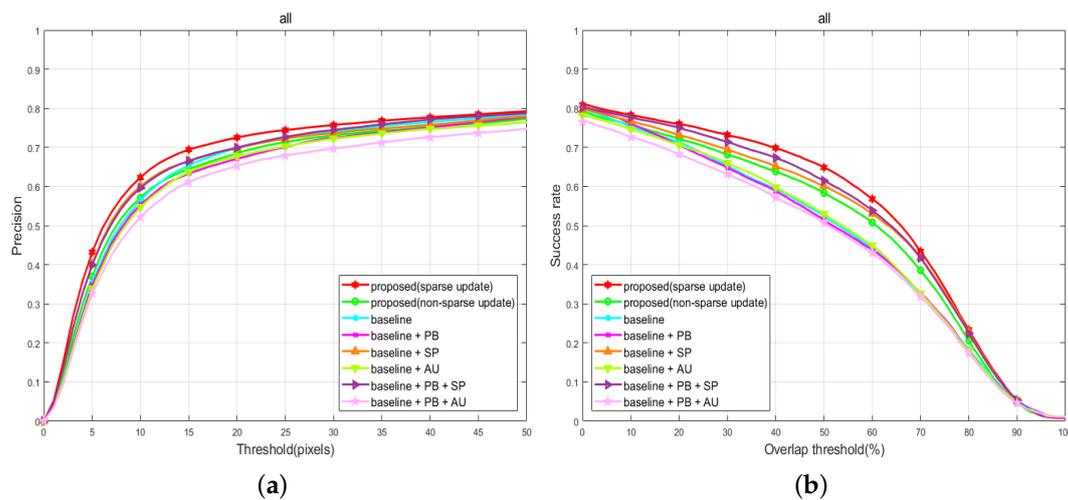
The precision plots and success plots for all experimental sequences with the proposed method and the state-of-the-art trackers are shown in Figure 5. The proposed method performed well across all thresholds, and the translation and scale estimation of the proposed method worked suitably. In particular, the success rate was generally higher than the precision, which means the estimated bounding box by our tracker was more overlapped with the bounding box from the ground truth.

In addition, an experiment was conducted to combine components in various manners using KCF as a baseline algorithm. Each component was the partial blocks (PB), scale pyramid (SP), adaptive update (AU) and sparse update (SU). Figure 6 shows the precision plots and success plots for each combination. Some combinations without considering scale variation such as PB and AU were detrimental to improving the tracking performance, and the proposed method combining all components performed better than any other combination.

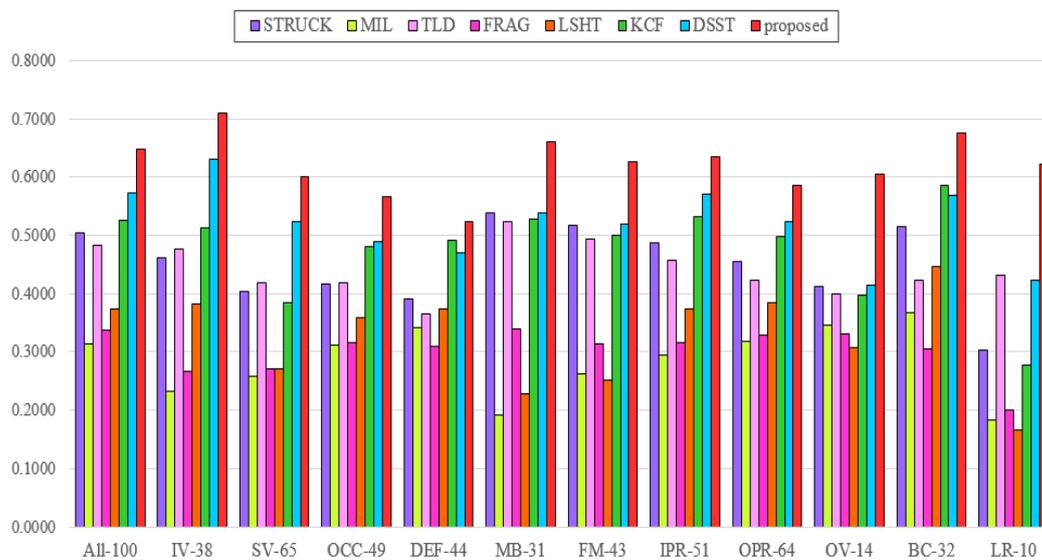
The OTB-100 contains sequences with attributes such as illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC) and low-resolution (LR). Figure 7 shows the average success rate of trackers for each attribute and demonstrates that the proposed method outperformed the existing trackers for all attributes.



**Figure 5.** The precision plots and success plots over all 100 sequences. The legend of the plots indicates the state-of-the-art trackers. (a) precision plots; (b) success plots.



**Figure 6.** The precision plots and success plots over all 100 sequences for each combination. (a) precision plots; (b) success plots.



**Figure 7.** The average success rate of the proposed tracker and existing trackers with all attribute sequences in OTB-100. The number indicates the number of sequences including attributes; illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC) and low-resolution (LR).

To consider the real-time tracker, we measured the frames-per-second (FPS) on the correlation filter-based trackers. KCF [21] and DSST [22] averaged 203 FPS and 43 FPS, respectively. The proposed method using the non-sparse updating version processed 35 frames per second. However, the sparse updating version processed 46 FPS on average for 100 sequences. As a result, the proposed method was slower than the KCF, but could be considered as a real-time tracker. Moreover, the proposed method outperformed the baseline KCF and DSST in terms of the AUC by 14.45% and 7.25%, respectively. Figure 8 shows the tracking results of the experimental sequences in OTB-100 database. From the top-left to bottom-right, the sequences in Figure 8 are Panda, Liquor, Freeman3, Walking2, Car1, Car24, Human8, Lemming, Box, Dog1, Coke, Vase, Skating1, CarScale, Singer1 and KiteSurf, respectively. The images contained various attributes, and the results of the proposed method were identified by a red bounding box. As intended, the proposed method was responsive to partial occlusion and scale variation.

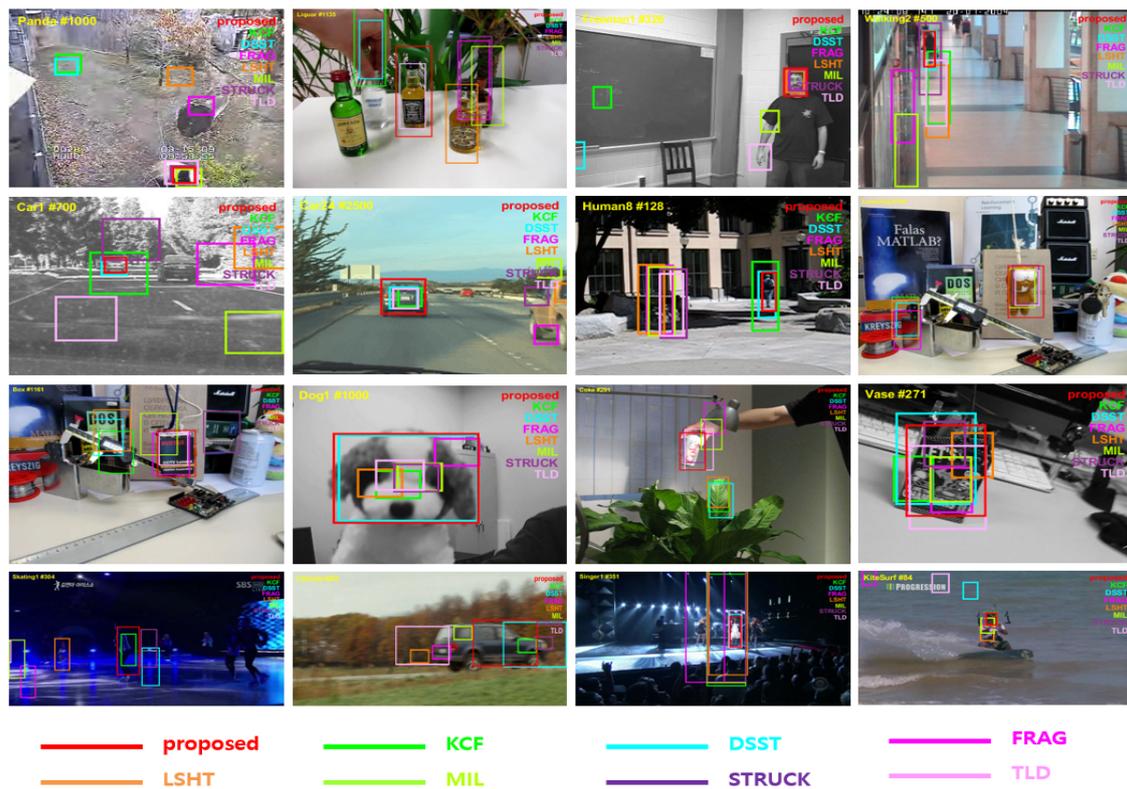


Figure 8. The experimental results comparison of the proposed tracker and existing trackers.

### 5. Conclusions

In this paper, we proposed a kernelized correlation filter-based visual object tracking algorithm using the partial block scheme and adaptive update model in the scale space. The proposed method accurately estimated translation and scale using the discriminative model. The proposed adaptive update model used the weighting function, which can be expressed as the combined sigmoid and gamma functions, to reduce the computational cost of calculating partial blocks for real-time tracking.

Various experiments were conducted to measure the performance of the trackers with the OTB-100 database. Experimental results validated that the proposed method outperformed existing state-of-the-art trackers in the sense of CLE, precision, success rate and AUC.

**Author Contributions:** S.J. designed the algorithm, performed the experiments and wrote the paper. J.P. supervised the research and reviewed the paper.

**Funding:** This research received no external funding.

**Acknowledgments:** This work was supported by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (2017-0-00250, Intelligent Defense Boundary Surveillance Technology Using Collaborative Reinforced Learning of Embedded Edge Camera and Image Analysis).

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

### References

1. Avidan, S. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2007**, *29*, 261–271. [[CrossRef](#)] [[PubMed](#)]
2. Grabner, H.; Grabner, M.; Bischof, H. Real-time tracking via on-line boosting. In Proceedings of the British Machine Vision Conference (BMVC), Edinburgh, UK, 4–7 September 2006; Volume 1.
3. Saffari, A.; Leistner, C.; Santner, J.; Godec, M.; Bischof, H. On-line random forests. In Proceedings of the 3rd IEEE ICCV Workshop on On-line Computer Vision, Kyoto, Japan, 27 September–4 October 2009.

4. Avidan, S. Support vector tracking. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2004**, *26*, 1064–1072. [[CrossRef](#)] [[PubMed](#)]
5. Hare, S.; Saffari, A.; Torr, P. Struck: Structured output tracking with kernels. In Proceedings of the IEEE Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
6. Babenko, B.; Yang, M.-H.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2011**, *33*, 1619–1632. [[CrossRef](#)] [[PubMed](#)]
7. Zhang, K.; Song, H. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognit.* **2013**, *46*, 397–411. [[CrossRef](#)]
8. Zhang, K.; Zhang, L.; Yang, M.-H. Real-time compressive tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012.
9. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2012**, *34*, 1409–1422. [[CrossRef](#)] [[PubMed](#)]
10. Collins, R.T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2005**, *27*, 1631–1643. [[CrossRef](#)] [[PubMed](#)]
11. Ross, D.A.; Lim, J.; Lin, R.-S.; Yang, M.-H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis. (IJCV)* **2007**, *77*, 125–141. [[CrossRef](#)]
12. Wang, D.; Lu, H. Visual tracking via probability continuous outlier model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 3478–3485.
13. Kwon, J.; Lee, K.M. Visual tracking decomposition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1269–1276.
14. Mei, X.; Ling, H. Robust visual tracking using L1 minimization. In Proceedings of the IEEE Conference on Computer Vision (ICCV), Kyoto, Japan, 27 September–4 October 2009; pp. 1436–1443.
15. Adam, A.; Rivlin, E.; Shimshoni, I. Robust fragments-based tracking using the integral histogram. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006.
16. Zhang, T.; Bibi, A.; Ghanem, B. In defense of sparse tracking: Circulant sparse tracker. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2016; pp. 3880–3888.
17. Zhang, T.; Ghanem, B.; Liu, S.; Ahuja, N. Robust visual tracking via multi-task sparse learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012; pp. 2042–2049.
18. Zhang, T.; Ghanem, B.; Liu, S.; Ahuja, N. Low-rank sparse learning for robust visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 470–484.
19. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
20. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012.
21. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2015**, *37*, 583–596. [[CrossRef](#)] [[PubMed](#)]
22. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
23. Danelljan, M.; Khan, F.S.; Felsberg, M.; van de Weijer, J. Adaptive color attributes for real-time visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
24. Shu, G.; Dehghan, A.; Oreifej, O.; Hand, E.; Shah, M. Part-based multiple-person tracking with partial occlusion handling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012; pp. 1815–1821.
25. Akin, O.; Mikolajczyk, K. Online Learning and Detection with Part-based Circulant Structure. In Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24 August 2014; pp. 4229–4233.

26. Yao, R.; Xia, S.; Shen, F.; Zhou, Y.; Niu, Q. Exploiting Spatial Structure from Parts for Adaptive Kernelized Correlation Filter Tracker. *IEEE Signal Process. Lett.* **2016**, *23*, 658–662. [[CrossRef](#)]
27. 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 (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 23–28.
28. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2016; pp. 1401–1409.
29. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 58–66.
30. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
31. Ruan, Y.; Wei, Z. Real-Time Visual Tracking through Fusion Features. *Sensors* **2016**, *16*, 949. [[CrossRef](#)] [[PubMed](#)]
32. Grabner, H.; Leistner, C.; Bischof, H. Semi-supervised on-line boosting for robust tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008.
33. Zhong, W.; Lu, H.; Yang, Mi. Robust object tracking via sparsity-based collaborative model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012.
34. Jia, X.; Lu, H.; Yang, Mi. Visual tracking via adaptive structural local sparse appearance model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012.
35. Sevilla-Lara, L.; Learned-Miller, E.G. Distribution fields for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012.
36. Oron, S.; Bar-Hillel, A.; Levi, D.; Avidan, S. Locally orderless tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012.
37. He, S.; Yang, Q.; Lau, R.; Wang, J.; Yang, Mi. Visual tracking via locality sensitive histograms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
38. Scholkopf, B.; Smola, A. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; The MIT Press: Cambridge, MA, USA, 2002.
39. Wu, Y.; Lim, J.; Yang, M. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
40. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.A.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2010**, *32*, 1627–1645. [[CrossRef](#)] [[PubMed](#)]



© 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/>).