Robust Visual Tracking Based on Fusional Multi-Correlation-Filters with a High-Confidence Judgement Mechanism

Wenbin Wang 1, Chao Liu 2,3, Bo Xu 2,3, Long Li 2,3, Wei Chen 1 and Yingzhong Tian 2,3,*

1 School of Mechanical and Electrical, Shenzhen Polytechnic, Shenzhen 518055, China; wangwenbin@szpt.edu.cn (W.W.); chenweil@szpt.edu.cn (W.C.)
2 Shanghai Key Laboratory of Intelligent Manufacturing and Robotics; liuchao_7523@shu.edu.cn (C.L.); wunian@shu.edu.cn (B.X.); lil@shu.edu.cn (L.L.)
3 School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China
* Correspondence: troytian@shu.edu.cn

Received: 3 January 2020; Accepted: 18 March 2020; Published: 21 March 2020

Abstract: Visual object trackers based on correlation filters have recently demonstrated substantial robustness to challenging conditions with variations in illumination and motion blur. Nonetheless, the models depend strongly on the spatial layout and are highly sensitive to deformation, scale, and occlusion. As presented and discussed in this paper, the colour attributes are combined due to their complementary characteristics to handle variations in shape well. In addition, a novel approach for robust scale estimation is proposed for mitigating the problems caused by fast motion and scale variations. Moreover, feedback from high-confidence tracking results was also utilized to prevent model corruption. The evaluation results for our tracker demonstrate that it performed outstandingly in terms of both precision and accuracy with enhancements of approximately 25% and 49%, respectively, in authoritative benchmarks compared to those for other popular correlation-filter-based trackers. Finally, the proposed tracker has demonstrated strong robustness, which has enabled online object tracking under various scenarios at a real-time frame rate of approximately 65 frames per second (FPS).

Keywords: Fusional correlation filters; visual tracking; high-confidence judge mechanism

1. Introduction

Robust visual object tracking has been attracting substantial attention. It is a significant problem in computer vision, as evidenced by its numerous implementations in robotics, services, monitoring, and human-machine interaction. Posada et al. [1] clearly defined computer vision as a key enabling technology for Industry 4.0. Segura et al. [2] and Posada et al. [3] showed the challenges and examples of computer vision technology in the field of robotics and human-robot collaboration. In typical scenarios, the target is specified in the first frame only (e.g. defining a rectangle), and it is often meaningful to track the target object in subsequent frames. This tracking can be directly applied in warehouse automation [4], human-robot handovers [5], safety design improvement for human-robot collaboration [4,6–9] and human-robot synchronization [10,11]. However, many challenges are encountered in visually tracking an object, which are due to challenging factors such as deformation, illumination variation, scale variation, and partial occlusions.

Most available methods that are used to solve visual tracking problems are based on two strategies: The first strategy is to use an efficient algorithm to construct generative [12–14] or discriminative [15–17] models. This strategy is commonly used to devise a filter or classifier for tracking the object and updating the model at each frame by utilizing the messages in subsequent frames as a training sample.
It might lead to model shift because a small error can accumulate into a significant error when learning from predictions. Primarily, this strategy is applied in scenarios with a lack of training samples. The second type is used to exploit the features that are extracted from a deep convolutional neural networks (CNN) [18–21], which is trained either online or on recognition datasets. Although these approaches can substantially improve the performance, the utilization of more complicated tracking algorithms or features would enormously increase the computational complexity, which might render the model unsuitable for real-time visual object tracking.

Recently, many popular trackers [21–24] that are based on correlation filters (CFs) have been proposed that can track many objects of interest because of their remarkable computational performance. By computing the correlation in the Fourier domain via the fast Fourier transform (FFT), the storage and computational requirements are both reduced by several orders of magnitude. Blome et al. [23] proposed a minimum output sum of squared error (MOSSE) visual object tracking method that uses adaptive correlation filters to incorporate a correlation filter, which can outperform more complicated algorithms. Due to the high performance and efficiency of CF, many trackers have been designed by adopting MOSSE. Henriques et al. [22] proposed the circulant structure of tracking-by-detection which extends the dense sampling that is based on MOSSE and introduces the kernel trick (CSK). In 2015, Henriques et al. [25] put forward a solution for high-speed tracking with kernelized correlation filters (KCFs), extended the multiple feature channels, and improved the performance of the tracker by utilizing the histogram of oriented gradients (HOG) feature, while preserving the real-time performance. Danelljan et al. [16] introduced the multiple feature channels of colour names (CNs), which are based on the CSK and have received an excellent response from the industry. However, although the trackers that are discussed above exhibit outstanding performance, they cannot solve the scale estimation problem due to the presence of fast motion or other factors. Long et al. [26] proposed an omnidirectional modified Laplacian operator with an adaptive window size. Danelljan et al. [27] (DSST) solved the difficult scale estimation problem by learning discriminative correlation filters that are adopted from the scale pyramid representation. Yingzhong et al. [28] used various fusion rules to combine different features for better description. For correlation filters, boundary effects might lead to detection failure; hence, Danelljan et al. [29] (SRDCF) added a spatial regularization term to penalize the CF coefficients around the boundary. The SRDCF yields excellent tracking results; however, the real-time performance is degraded enormously, with a reported speed of only 5 FPS. In the development of trackers that are based on CF, the discrimination performance should be improved and the real-time performance requirement should be satisfied.

Due to their strong feature representation performances, CNNs have realized significant success on visual tracking tasks and in many other scenarios. Many recent trackers [18–21,30–32] have demonstrated high-performance on benchmarks. Lee et al. [32] and Wang et al. [31] proposed the best-performing solutions for the visual object test (VOT) [33] long-term tracking and short-term tracking, respectively. Ma et al. [30] proposed a method for enhancing the precision and robustness by learning the features that are extracted from deep convolutional neural networks and a CNN that is trained on object recognition datasets. Danelljan et al. [21] exploited an implicit interpolation model with the objective of solving the learning problem in the continuous spatial domain and proposed an innovative method for efficiently combining multi-resolution feature maps. Nam et al. [20] exploited a network by incorporating domain-specific layers and shared layers to obtain generic target representations. Wenbin et al. [34] proposed a generative system that is based on CNN, which has realized satisfactory performance. Yang Liu et al. [35] proposed a novel hierarchical feature learning framework and Dongdong et al. [36] revisited the standard SRDCF formulation and introduced padless correlation filters (PCFs), which could completely remove boundary effects. The studies that are discussed above have demonstrated the high power of CNN for target representation at the expense of high computational complexity and time consumption.

In recent years, the practice of evaluating tracking algorithms has substantially improved. In the past, researchers were limited to evaluating the tracking performance of a small number
of sequences [37,38]. Benchmarks such as VOT, and the object tracking benchmark (OTB) [39,40] emphasize the importance of test methods for a wider range of sequence sets that cover a variety of object categories and challenges. OTB contains 25% grayscale sequences, while VOT contains only colour sequences. OTB includes the start of a random frame and is initialized via the addition of random interference, whereas VOT is initialized and run from the first frame.

In this paper, an understandable and efficient method is proposed for solving the problem that is described above. The main contributions can be summarized as follows: Two image representations of template and colour characteristics are combined to address illumination changes and shape variations. The discriminative CF is exploited based on a scale pyramid representation to solve the scale estimation problem. A high-confidence judgement mechanism is explored for avoiding model corruption. Figure 1 shows the flow of the target tracking algorithm in this paper.

Figure 1. Procedure for Our Approach. In Figure 1, the blue line is template-related. In frame $t$, the HOG features are extracted from the estimated location and used to update the numerator $A_t$ and denominator $B_t$ in (14). In frame $t+1$, features are extracted from the predicted target location $p_t$ and convolved with $H_t$ to obtain the template response via (15). The green line is histogram-related. In frame $t$, features of the target object and background are used to update $\rho_t(O)$ and $\rho_t(B)$ in (20); thus, the coefficient $\beta_t$ in (19) can be obtained. The histogram response is computed via (5). Then, the template response and the histogram response are combined via (3) to obtain the integrated response map without adjusting the scale. The orange line is scale-related. In frame $t$, a scale-space filter is trained by using the feature from previous location $p_t$. In frame $t+1$, the scale filter is combined with the response from (3) to obtain the final response $p_{t+1}$. The red line is high-confidence judgement-related. The high-confidence mechanisms in (21) and (22) are used to judge whether updating model in the current frame is necessary for the prevention of model corruption.

2. Combining Colour Characteristics

2.1. Problem Formulation

In this paper, the detection principle is utilized for tracking. The main objective is to obtain a classifier that can discriminate the object of interest from its ambient environment in real time when a new frame is received. In frame $t$, the rectangle $p_t$ represents the object position in picture $x_t$ which is selected from a collection $C_t$ to maximize a fraction:

$$p_t = \arg \max_{p \in C_t} f(T(x_t, p); \theta_t-1)$$  \hspace{1cm} (1)
where \( f(T(x, p); \theta) \) denotes a fraction of the rectangular window \( p \) in picture \( x \) with the parameters \( \theta \) of the model, and the function \( T \) denotes an image transformation. Moreover, the parameters of the model should be selected to minimize the loss function \( L(\theta; \chi_t) \) based on the foregoing pictures and the positions of the target object in these pictures \( \chi_t = \{(x_t, p_t)\}_{t=1}^T \):

\[
\theta_t = \arg \min_{\theta \in \Omega} \{ L(\theta; \chi_t) + \gamma R(\theta) \}
\]

(2)

The space of model parameters \( \theta \) is represented by \( \Omega \). The regularization term \( R(\theta) \) with weight coefficient is \( \gamma \) used to restrict the model complexity and to avoid over-fitting. To realize real-time performance, the problems in (1) and (2) should be covered well, and the functions \( f \) and \( L \) should be selected to render the location of the target object reliable and accurate.

A fraction function is proposed which is a linear composition of histogram and template fractions in which the template fraction is obtained from the HOG feature and the histogram fraction is obtained from the CN feature:

\[
f(x) = \gamma_{tmpl} f_{tmpl}(x) + \gamma_{hist} f_{hist}(x)
\]

(3)

The template fraction is a linear function of a N-channel feature picture \( \varphi_x : \Gamma \to \mathbb{R}^N \), that is acquired from \( x \) and acts on a finite area \( \Gamma \subset \mathbb{Z}^2 \). The weight vector \( a \) is another N-channel image:

\[
f_{tmpl}(x; a) = \sum_{u \in \Gamma} a[u]^T \varphi_x[u]
\]

(4)

The histogram fraction is obtained from an M-channel feature picture \( \varphi_x : H \to \mathbb{R}^M \) that originates from \( x \) and acts on a finite area \( H \subset \mathbb{Z}^2 \):

\[
f_{hist}(x; \beta) = g(\varphi_x; \beta)
\]

(5)

In contrast to the template fraction, the histogram fraction is invariant to the spatial arrangement because the proportion of an object colour distribution is relatively constant. A linear function is used to represent the average feature pixel:

\[
g(\varphi_x; \beta) = \beta^T \left( \frac{1}{|H|} \sum_{u \in H} \varphi_x[u] \right)
\]

(6)

It can be expressed as the average of a score image \( \zeta_{(\beta; \psi)}[u] = \beta^T \psi[u] \):

\[
g(\varphi_x; \beta) = \frac{1}{|H|} \sum_{u \in H} \zeta_{(\beta; \psi)}[u]
\]

(7)

It is significant to transform the feature by applying the translation \( \varphi_{T(x)} = T(\varphi_x) \) such that the computation of the feature can be conducted by using overlapped windows, and the template fraction can be calculated by adopting the fast approaches that are commonly used for convolution processes. By using a single integral image, the histogram score is obtained.

The parameters of the whole model are \( \theta = (a, \beta) \), and the coefficients \( \gamma_{tmpl} \) and \( \gamma_{hist} \) can be inferred from \( a \) and \( \beta \). The loss function \( L(\theta; \chi) \) should be optimized by adjusting the parameters to obtain a weight-based linear composition of per-picture losses:

\[
L(\theta; \chi_T) = \sum_{t=1}^{T} w_l(x_t, p_t, \theta)
\]

(8)

The form of the per-picture loss function is as follows:

\[
l(x, p, \theta) = \Delta d(p, \arg \max_{p \in C} f(T(x, q); \theta))
\]

(9)

where \( \Delta d(p, q) \) represents the cost of selecting the rectangle \( q \) while the true rectangle is \( p \). Since it is a non-convex function, the computation of the optimization problem is exceedingly expensive, and the
quantity of training specimen and features are limited. In contrast, correlation filters, adopt a simplistic least-squares loss function and many specimens are created by using cyclic shifts. Moreover, all the circular matrices could be diagonalized to reduce the amount of computation substantially by using the discrete fourier transform (DFT).

To maintain the efficiency and performance of the correlation filter without losing sight of the message that can be acquired from a permutation-invariant fraction of the histogram, construction of the model by solving settling two absolute ridge regression problems is suggested:

\[ \alpha_t = \arg \min \left\{ L_{tmpl}(\alpha; \chi_t) + \frac{1}{2} \lambda_{tmpl} \| \alpha \|^2 \right\} \]  

(10)

\[ \beta_t = \arg \min \left\{ L_{hist}(\beta; \chi_t) + \frac{1}{2} \lambda_{hist} \| \beta \|^2 \right\} \]  

(11)

By applying the correlation filter formulation, the parameters \( \alpha \) can be easily obtained. Although the dimension of \( \beta \) may be less than \( \alpha \), it could still be more difficult to determine. This is because it is not possible to acquire it from the circular shifts. It must be converted to a common matrix instead of a circular matrix.

In the end, the linear composition of the two fractions is used to make \( \gamma_{hist} = \varepsilon \) and \( \gamma_{tmpl} = 1 - \varepsilon \), where \( \varepsilon \) is a coefficient that was selected from a validation set.

2.2. Obtaining the Template Fraction

According to a least-squares expression of the correlation filter, the loss function should be:

\[ l_{tmpl}(x, p, h) = \left\| \sum_{l=1}^{d} h^l \ast f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \| h^l \|^2 \]  

(12)

where \( h^l \) represents the channel \( l \) of the multi-channel image \( h \), and \( g \) is the expected correlation result that corresponds to the training sample \( f \). The regularization parameter \( \lambda \) is used to restrict the model computation and to avoid the problem of zero-frequency components in the spectrum \( f \). It is supposed that (12) only has one training sample so that the solution to (12) can be obtained.

\[ H^l = \frac{\overline{G}^l}{\sum_{k=1}^{d} F^k F^k + \lambda} \]  

(13)

Here, \( \overline{G} \) denotes the complex conjugate of \( G \) which should be equal to \( \overline{F} \). The filter can be optimized by minimizing the output error on all training samples, but the computation for solving this problem is enormous. To obtain the efficient and convenient approximation, the numerator \( A_i \) and denominator \( B_i \) of the correlation filter \( H^l \) in (13) are independently updated as follows:

\[ A_i^l = (1 - \eta)A_{i-1}^l + \eta \overline{G} F^l, \quad l = 1, \ldots, d \]  

(14a)

\[ B_l = (1 - \eta)B_{l-1} + \eta \sum_{k=1}^{d} \overline{F}^k F^k \]  

(14b)

Here, \( \eta \) is a parameter of the learning rate. The formula that is used to compute the correlation of the fractions \( y \) on the rectangular area \( z \) of a feature map is (15). The new target position can be found by maximizing the score \( y \), and \( g^{-1} \) denotes the inverse DFT operator.

\[ y = g^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A}^l z^l}{B + \lambda} \right\} \]  

(15)
2.3. Obtaining the Histogram Fraction

The histogram fraction can be obtained from the specimens that are acquired from each picture, where \( W \) represents a collection of pairs \((q, g)\) of the rectangular window \( q \) and their related regression output \( g \in \mathbb{R} \), and the loss function is:

\[
l_{\text{hist}}(x, p, \beta) = \sum_{(q, g) \in W} \left( \beta^T \left[ \sum_{u \in H} \psi_T(x, u) \right] - g \right)^2
\]

(16)

For an N-channel feature that can transform \( \psi \), the answer can be acquired by solving a \( N \times N \) system of equations, which consuming \( O(N^2) \) memory and \( O(N^3) \) time. This is a challenging task to complete if the number of features is enormous.

Instead, features of the form \( \psi[u] = e_k[u] \) are put forward, where \( e_i \) is a one-hot vector that is one at index \( i \) and zero at other places. Moreover, \( \beta^T \psi[u] = \beta^k[u] \) is utilized in the PLT method, which is demonstrated in [41]. The type of features is selected as RGB colours, even though the local binary patterns would be a suitable alternative. To render this approach more efficient and convenient, linear regression is conducted on every feature pixel of the target object, and background areas \( O \) and \( B \subset \mathbb{Z}^2 \) and the per-picture loss function is transformed to:

\[
l_{\text{hist}}(x, p, \beta) = \frac{1}{|O|} \sum_{u \in O} \left( \beta^T \psi[u] - 1 \right)^2 + \frac{1}{|B|} \sum_{u \in B} \left( \beta^T \psi[u] \right)^2
\]

(17)

Here, \( \psi \) abbreviates of \( \psi_T(x, p) \). By adding the one-hot encoding, the formula above can be transformed into the independent parts according to the feature dimension:

\[
l_{\text{hist}}(x, p, \beta) = \sum_{j=1}^{N} \left[ \frac{M^j(O)}{|O|} \left( \beta^j - 1 \right)^2 + \frac{M^j(B)}{|B|} \left( \beta^j \right)^2 \right]
\]

(18)

Here, \( M^j(A) = |\{ u \in A : k[u] = j \}| \) is the number of pixels in the area \( A \) of \( \psi_T(x, p) \) where the feature \( j \) is non-zero and \( k[u] = j \). Then, the corresponding ridge regression problem is:

\[
\beta^j = \frac{\rho^j(O)}{\rho^j(O) + \rho^j(B) + \lambda}
\]

(19)

Here, for every feature dimension \( j = 1, \ldots, N \), \( \rho^j(A) = M^j(A) / |A| \) is the ratio of the number of pixels in the area where feature \( j \) is non-zero. The parameters of the model are updated online

\[
\rho_t(O) = (1 - \eta_{\text{hist}}) \rho_{t-1}(O) + \eta_{\text{hist}} \rho^j_t(O)
\]

(20a)

\[
\rho_t(B) = (1 - \eta_{\text{hist}}) \rho_{t-1}(B) + \eta_{\text{hist}} \rho^j_t(B)
\]

(20b)

3. Combining the Scale Space Filter

In this chapter, a combined colour attribute that is used for translation in the main approach is proposed for alleviating the effects of deformation. Afterward, a solution is proposed for effectively overcoming the problem of scale estimation. In contrast to the traditional scale estimation approaches, an advanced approach of adaptive scale estimation that is based on the established object location was proposed for avoiding the high complexity that is caused by exhaustive search.

3.1. Combining the Scale Space Filter

A convenient and simple strategy is proposed for incorporating the scale estimate of a 3-dimensional scale-space filter. Via this approach, the translation and scale can be estimated together by computing the fractions in the area where the shape is similar to a box of scale pyramid manifestations, and the scale can be estimated by maximizing this fraction.
First, a feature pyramid around the specified target location in a rectangular region is constructed. The feature pyramid is constructed such that the target size in the region that corresponds to the spatial filter has $M \times N$ dimensions. The training specimen $f_t$ is a rectangular cube of dimensions $M \times N \times S$ and is located around the object position and scale, where $S$ represents the magnitude of the scale-space filter. The filter is renovated via (14), and the expected correlation output $g$ can be obtained by using a 3-dimensional Gaussian function. The building of the training samples is illustrated in Figure 2.

![Figure 2. The Visualization of Scale Space Filter Samples.](image)

Figure 2 shows a visualization of the scale space filter samples that are extracted from the feature pyramid. The left image represents the feature pyramid which is built up around the centre of the target object, where $S$ is the number of plies of the feature pyramid, and $d$ is the number of dimensions of each sample. The right image shows that samples of various sizes among layers are extracted to make the training more powerful.

During the period of detection, a feature pyramid is constructed based on the preceding target location and the scale of the main estimation approach. The rectangular cube of size $M \times N \times S$ that is located around this position is applied as the test specimen that corresponds to $z$ in (15). Afterward, the correlation scores $y$ can be computed via Equation (15) (2.2).

### 3.2. Iterative Scale Space Filter

As described in Section 3.1, the feature pyramid is constructed centred on the previously estimated target location and scale. This might cause the inclusion of a shearing ingredient in the conversion about the test sample $z$.

The effects of the scale shearing distortions can be alleviated by iterating the detection procedure of the tracker model. Thus, a joint scale-space filter can be iteratively adopted. When accepting a new frame, the filter of the preceding object scale and location is first utilized.

Afterward, the current object location is renovated with the scale and location when the transformation correlation filter and scale-space filter attain the maximum scores separately. Then, the detection procedure is iterated by building the feature pyramid around the current object estimate. The process always converges due to the alleviated shearing distortion, which is regarded as a parameter, when the accuracy of location estimation is enhanced.

### 4. High-Confidence Judgement Mechanism

The model update strategy significantly impacts the precision and robustness of the tracking algorithm because the appearance of the target object varies in the tracking scenario. However, the model that is used to detect the position in the current frame is unchanged; hence, the object information that has changed cannot be obtained. Thus, a model update strategy is necessary. However, if the update frequency is too high, the problems of occlusion and motion blur cannot be solved effectively, and if the frequency is too low, the model cannot timely learn the new feature from
the ambient environment. To overcome these challenges, a high-confidence judgement mechanism is proposed.

In Figure 3, the left column contains two frames of sequence \textit{basketball} from OTB-15, the red bounding boxes represent the tracking results of our algorithm with the high-confidence judgement mechanism, and the blue bounding boxes represents the tracking results that are obtained when the approached judgement mechanism is not utilized and the method of updating the model is adopted in each frame. The middle column describes the scenario of severe occlusion of the target, under which the model cannot be updated. The right column presents the map response to updating the model in the same scenario.

![Figure 3. The effect of high-confidence judge mechanism.](image)

Most current trackers renovate tracking models at each frame without considering the detection precision. This might leads to deterministic failure if inaccurate detection, severe occlusion or complete object absence occurs. In this section, the response from the tracking results is utilized to judge whether it is necessary to renovate the tracking model or not.

The peak fraction and the volatility level of the response cartographic represent the confidence level of the tracking outputs.

The desired response cartographic should only have readily observable peaks and should be smooth in all other regions if the tracking outputs are specially matched to the accurate target location and scale. The more readily observable the correlation peaks are, the higher the location precision is.

Otherwise, as shown in the first row of Figure 3, the response map will fluctuate violently in which the representation differs entirely from general response maps. If the tracking model is continuously updated when a deterministic failure occurs, it should be erroneous, as shown Figure 3 in the second row. Therefore, a high-confidence judgement mechanism is proposed for avoiding this scenario, which considers two standards: The first standard is the maximum response fraction \( f_{\text{max}} \) of the response cartographic \( f(x, p; \theta) \):

\[
  f_{\text{max}} = \max f(x, p; \theta)
\]  

(21)

The second standard, namely, the average fluctuate-except-peak energy (AFEPE) is a novel standard that is used to express the volatility level of response cartographic and the confidence degree of the detected targets which is defined as follows:

\[
  AFEPE = \frac{|f_{\text{max}} - f_{\text{min}}|^2}{\text{avg} \left( \sum_{w,h} (f_{w,h} - f_{\text{min}})^2 \right)}
\]  

(22)

Here \( f_{\text{max}} \) represents the maximum, \( f_{\text{min}} \) represents the minimum, and \( f_{w,h} \) represents the \( w \)-th row and \( h \)-th column elements of \( f(x, p; \theta) \).
In the ideal scenario, in which the complete target appears in the detection scope, AFEPE should exceed the cartographic response. It should have only a single readily observable peak and should be smooth in all other regions. In contrast, AFEPE will dramatically decrease when the target is disappearing or occluded.

While the two standards, namely, \( f_{\text{max}} \) and AFEPE, of the current frame are larger than the historical mean values that are used separately in various proportions, the tracking results in this frame are high-confidence result. Then, the tracking model will be updated online via (14), (20), and (21).

Figure 3 shows the main advantage of the proposed method. While the object is occluded severely, the response cartographic waves intensify and AFEPE decreases to approximately 10 when \( f_{\text{max}} \) is still sufficiently large. In this scenario, the high-confidence judgement mechanism would not renovate the model. With this approach, the tracking model will not be corrupted and the model can track the target object successfully once again in the subsequent tracking. Otherwise, the object will disappear, and our desired peak may vanish gradually.

5. Experiments

To evaluate the performance of our proposed tracking algorithm, we conduct an experiment on the OTB-13 [39] and OTB-15 [40] benchmark datasets. OTB is an authoritative benchmark that is used by many visual tracking researchers to evaluated the feasibility and efficiency of their proposed approaches. The evaluation methods for OTB-13 and OTB-15 are the same. OTB-13 has 50 sequences, while OTB-15 has 100 sequences. These sequences differ in terms of their challenging conditions; hence, they can be used to evaluate trackers more comprehensively.

All test sequences of OTB have been tagged with 11 attributes which represent challenging conditions in various scenarios, such as background clutters (BC), motion blur (MB), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out-of-plane rotation (OPR), scale variation (SV), deformation (DEF), fast motion (FM) and out-of-view (OV). Using these 11 metrics, we evaluate the performance of our approach under various scenarios. Furthermore, we utilize the metrics of OPE (one-pass evaluation) and SRE (spatial robustness evaluation). The success scores is the area under the curve (AUC), which describes the success rate when the estimated positions compare to the ground-truth positions with a fixed overlap threshold that ranges from 0 to 1. The precision scores mean is the percentage of the estimated centre positions that are within 20 pixels of the ground-truth centre positions.

In this paper, we compare our algorithm with 7 state-of-the-art high-performance trackers on OTB-13 and OTB-15. We use the experimental results of 7 trackers that have been published by their authors to guarantee a fair comparison. Our algorithm is represented by Algorithm 1.

In Table 1, we list important parameters that we used in our experiments. The values of these parameters is selected by conducting experiments with various parameter values to select the optimal values, which renders our approach more powerful. Our tracker is implemented in MATLAB on a Notebook with an Intel i5-6200U @2.3GHz processor.

| Parameters                  | Value |
|-----------------------------|-------|
| Learning rate(template) \( \eta_{\text{tmpl}} \) | 0.02  |
| Learning rate(histogram) \( \eta_{\text{hist}} \) | 0.05  |
| Merge factor(template) \( \gamma_{\text{tmpl}} \) | 0.70  |
| Merge factor(histogram) \( \gamma_{\text{hist}} \) | 0.30  |
| Colour features             | RGB   |
**Algorithm 1** Proposed tracking approach: start from step \( t \).

**Input:**
1. Frame \( x_t \). Previous target object position \( p_{t-1} \) and scale \( s_{t-1} \). Previous template model \( A_{t-1}^{tmpl}, B_{t-1}^{tmpl} \), histogram model \( \rho_{t-1}(O), \rho_{t-1}(B) \) and scale model \( A_{t-1}^{scale}, B_{t-1}^{scale} \).

**Output:**
2. Estimated target position \( p_t \) and scale \( s_t \). Updated template model \( A_t^{tmpl}, B_t^{tmpl} \), histogram model \( \rho_t(O), \rho_t(B) \) and scale estimation model \( A_t^{scale}, B_t^{scale} \).

3. repeat
   4. **Translation estimation:**
      5. Extract translation samples \( z_{trans} \) from \( x_t \) at \( p_{t-1} \) and \( s_{t-1} \).
      6. Get the translation correlation \( f(x_t) \) using \( z_{trans} \) with \( A_{t-1}^{tmpl}, B_{t-1}^{tmpl} \) and \( \rho_{t-1}(O), \rho_{t-1}(B) \) in Equation (4)(2.1), Equation (5)(2.1) and combined these in Equation (4)(2.1).
      7. Set \( p_t \) to the target position in the current frame that maximizes \( f(x_t) \).
      8. **Scale estimation:**
         9. Extract scale samples \( z_{scale} \) from \( x_t \) at \( p_{t-1} \) and \( s_{t-1} \).
         10. Get the scale correlation \( y_{scale} \) using \( z_{scale}, A_{t-1}^{scale}, B_{t-1}^{scale} \) in Equation (15)(2.2).
         11. Set \( s_t \) to the target scale that maximizes \( y_{scale} \).
      12. **Model update:**
         13. Compute \( f_{max} \) and AFEPE in Equation (21)(4) and Equation (22)(4).
         14. if \( f_{max} \) and AFEPE larger than historical mean values then
         15. Update the translation model \( A_t^{tmpl}, B_t^{tmpl} \) and \( \rho_t(O), \rho_t(B) \) in Equation (14)(14b) and Equation (20)(20b).
         16. Update the scale model \( A_t^{scale}, B_t^{scale} \) in Equation (14)(14b).
         17. end if
      18. until end of the tracking sequences.

### 5.1. Analyses of Our Approach

To evaluate the performance of our proposed approach with three strategies, we evaluate four versions of our algorithm on OTB-13: Ours-NHCN, in which colour names are not combined; Ours-NSP, in which the scale space filter is not utilized; Ours-NHCM, which lacks the high-confidence judgement mechanism; and Ours-N3, in which none of the three strategies are utilized. The characteristics and comparison results are presented in Table 2.

Comparing the trackers of Ours and Ours-NHCM, precision and success have improved by 6.8% and 13.0%. In addition, FPS has increased by 22.6%. Thus, the high confidence judgement mechanism can alleviate the model corruption that is caused by occlusion and other problems, and the strategy of updating when necessary instead of updating every frame has substantially improved the speed.

Without combining the colour features, the tracker Ours-NHCN shows poorer performance. Compared with Ours-NCN, the precision of Ours increased by 15.7% and the success rate increased by 30.5%, but at the expense of 11 FPS. Ours-N3, as might be expected, performs worst due to the absence of the benefits of the three strategies. Moreover, its speed is the fastest due to its lower complexity.

According to Table 2, tracker Ours outperforms all the other versions in terms of precision and success rate, and the speed can reach 65 FPS, which satisfies real-time requirements. As discussed above, the three mechanisms that are proposed in this paper realize satisfactory efficiency and feasibility.
Table 2. The characteristics and comparison results of different versions of our approach on OTB-13.

| Trackers | Colour attributes | Scale Space Filter | High-Confidence Mechanism | Precision | Success | FPS |
|----------|-------------------|--------------------|---------------------------|-----------|---------|-----|
| Ours-N3  | No                | No                 | No                        | 0.682     | 0.458   | 85  |
| Ours-NCN | No                | Yes                | Yes                       | 0.705     | 0.512   | 76  |
| Ours-NSP | Yes               | No                 | Yes                       | 0.746     | 0.577   | 69  |
| Ours-NHCM | Yes              | Yes                | No                        | 0.764     | 0.591   | 53  |
| Ours     | Yes               | Yes                | Yes                       | 0.816     | 0.668   | 65  |

5.2. Overall Performance Evaluation

In this section, we evaluate the performances of the various versions of our approach. For further comparison, we evaluate our algorithm with 7 state-of-the-art trackers, namely, Staple [24], fDSST [19], KCF [25], CN [16], CSK [22], GOTURN [42] and CCOT [43], as listed in Table 3. Among them, Staple, fDSST, KCF, CN, and CSK are correlation-filter-based algorithms, and GOTURN and CCOT are deep-learning-based algorithms. The data of GOTURN and CCOT are obtained from their original papers.

Table 3. The comparison results with the other 7 state-of-the-art trackers on OTB-13 and OTB-15.

| Trackers | Correlation Filters | Deep Learning | OTB-13 Precision | OTB-13 Success | OTB-15 Precision | OTB-13 Success | FPS |
|----------|---------------------|---------------|------------------|----------------|------------------|----------------|-----|
| Staple   | Yes                 | No            | 0.767            | 0.617          | 0.769            | 0.594          | 80  |
| KCF      | Yes                 | No            | 0.703            | 0.456          | 0.710            | 0.438          | 172 |
| CN       | Yes                 | No            | 0.637            | 0.432          | 0.641            | 0.411          | 78  |
| CSK      | Yes                 | No            | 0.490            | 0.324          | 0.517            | 0.305          | 151 |
| fDSST    | Yes                 | No            | 0.696            | 0.526          | 0.705            | 0.512          | 54  |
| GOTURN   | No                  | Yes           | 0.620            | 0.444          | 0.572            | 0.427          | 165 |
| CCOT     | No                  | Yes           | 0.899            | 0.672          | -                | 0.682          | 0.3 |
| Ours     | Yes                 | No            | 0.816            | 0.668          | 0.802            | 0.646          | 65  |

According to Table 3, CCOT outperforms all the other trackers, and compared to our approach, it realizes average improvements of approximately 10.2% in precision on OTB-13 and 5.6% in success on OTB-15. They perform almost the same in terms of success on OTB-13, and the speed of CCOT is only 0.3 FPS, which cannot satisfy the real-time requirements. In contrast, the speed of our algorithm is 65 FPS, which runs more than 216 times faster than CCOT. Moreover, tracker GOTURN, which is also based on deep learning, runs very fast at 165 FPS and is second only to KCF in terms of speed; however, a large disparity in performance is observed compared with our tracker. Our approach realizes average enhancements of approximately 25% in precision and 49% in success.

Figure 4 presents the precision and success plots of the top six trackers on both OTB-13 and OTB-15. The first row in Figure 4 presents the comparison results on OTB-13, and the second row in Figure 4 presents the comparison results on OTB-15. These six trackers are all correlation-filter-based algorithms, and their characteristics are presented in detail in Table 4. KCF and CSK only use HOG features to describe the object model, and they do not utilize the three strategies. CN uses colour names to avoid object deformation problems, and fDSST focuses mainly on scale estimation; their speeds are 78 FPS and 54 FPS, respectively. Our tracker fully utilizes the three mechanisms and performs best in terms of most metrics.
Figure 4. The precision and success plots of OPE on OTB-13 and OTB-15.

CSK is a famous tracker that pioneered the application of correlation filters (CFs) in visual tracking; hence, it is a satisfactory representation of previous classical trackers. Our approach significantly improves CSK, with an average improvement of approximately 60% in precision. KCF adopts cyclic shifts for dense sampling and joins multi-channel HOG features to make the tracker more robust. It can run very fast at 172 FPS. Our tracker significantly outperforms KCF, with average enhancements of the success rate of approximately 46.5% on OTB-13 and 47% on OTB-15.

Table 4. The characteristics of 6 trackers based on correlation filters.

| Trackers | Colour Attributes | Scale Space Filter | High-Confidence Mechanism |
|----------|-------------------|--------------------|--------------------------|
| Staple   | Yes               | Yes                | No                       |
| KCF      | No                | No                 | No                       |
| CN       | Yes               | No                 | No                       |
| CSK      | No                | No                 | No                       |
| fDDST    | No                | Yes                | No                       |
| Ours     | Yes               | Yes                | Yes                      |

CN utilizes the feature of multi-channel colour names that is based on CSK, and fDSST is the speedy version of DSST with a scale space filter. They have realized satisfactory improvements, but without the colour names and high-confidence judgement mechanism, the performance is worse than that of our approach. By combining HOG features and colour names, Staple realizes relatively satisfactory performance. Moreover, on the basis of multi-feature fusion, our tracker employs adaptive scale estimation and a high-confidence judgement mechanism. Our tracker outperforms Staple by 5.3% in terms of precision and 8.5% in terms of the success rate on average.

5.3. Robustness Evaluation

Figure 5 presents a visualization of the tracking results of our tracker and of other famous trackers on various test sequences. The “Liquor”, “Skating”, “Jogging-2” and “Subway” sequences all contain occlusions, deformations, and background clutter, which lead CSK and KCF to miss the object completely, whereas our tracker can track the object accurately.

In sequences “Skating” and “Jogging-2” which contain illumination variations, scale variations, occlusions, deformations, out-of-plane rotations, and background clutter, all trackers identify the target...
in the first several frames, but most trackers lose sight of the object over time, and only our approach can always track the object correctly.

**Figure 5.** The visualization of comparison results of our proposed tracker and other existing trackers on five challenging sequences, i.e, (a) Skating, (b) Couple, (c) Subway, (d) Jogging-2, (e) Liquor.

Figure 6 presents the success plots on 9 challenging scenarios: fast motion (FM), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out-of-plane rotation (OPR) and scale variation (SV). Our proposed tracker performs best in most scenarios.

We can see that the success rates in the scenarios of background clutter, motion blur, deformation and illumination variation in (b), (c), and (d), respectively, are approximately twice those of the original correlation filter that is obtained by combining colour attributes. In (i), our approach realizes a larger improvement by approximately 14.2% compared to Staple, which does not utilize adaptive scale estimation. In (g), it demonstrates strong performance, which is due to the effect of the high-confidence judgement mechanism when encountering update problems.

According to these experiments, our proposed fusional multi-correlation -filters with the high-confidence judgement mechanism can outperform state-of-the-art trackers in most scenarios.
5.4. Experiment with the Target Tracking Robot

Target tracking robots are potential applications of human-robot collaboration and can be applied to service areas, such as industrial plants, offices, supermarkets, airports, etc. and collaboratively provide information, guidance and/or physical assistance to people. The main challenge for the target tracking robot is to track the user accurately and in real-time.

The robot platform used in this experiment was a GQY robot. GQY robot is a comprehensive experimental platform, which is equipped with a monocular camera, lidar, ultrasonic sensor, and other sensors. We used the GQY robot to experiment with our tracker.

If the robot cannot track the experimenter in real-time, it will not follow the experimenter in an S-shaped path. Figure 7 shows the target tracking robot following the experimenter in an S-shaped path. We repeated the experiment more than 20 times. Thanks to the real-time improvement of our tracker, the target tracking robot can track the target stably in real-time.

As shown in Figure 8, the illumination variation during the process is very obvious. Our tracker, which incorporates colour attributes, is more robust in scenes with illumination variation, and illumination variation does not affect the accuracy and stability of robot tracking.

In a variety of challenging environments, the target tracking robot can still perform effective and accurate tracking, which illustrates the effectiveness of our tracker and can meet the needs of the target tracking for real-time target tracking.
6. Conclusions

In this paper, a novel tracker is proposed for overcoming challenges such as deformation, scale variation, and occlusion in the field of visual tracking. The colour attributes are combined, and due to their complementary characteristics, they are used to well handle variations in shape. Correlation-filter-based trackers have theoretical limitations. For example, large shape changes can lead to more background be learned, which affects the correlation-filter-based trackers. To mitigate the problems that are caused by background clutters, fast motion and scale variations, an innovative scale estimation filter is utilized. Furthermore, a high-confidence mechanism is proposed for preventing model corruption. This tracker not only performs excellently but also satisfies the real-time performance requirement of online object tracking. Deep learning has advantages in feature representation. The trained network can achieve high accuracy. The target tracking algorithm proposed in this paper is based on correlation filtering and ensures the real-time performance of the algorithm; however, its accuracy is lower than that of deep-learning-based target tracking algorithms. We are combining our method with deep learning to further increase accuracy.

Author Contributions: In this work, W.W., C.L. and B.X. conceived the main idea, designed the main algorithms, experiments and wrote the paper. L.L., Y.T., and W.C. analyzed the data, performed the simulation experiments and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Special Plan of Major Scientific Instruments and Equipment of the State (Grant No.2018YFF01013101), National Natural Science Foundation of China (51775322, 61603237), Project named “Key technology research and demonstration line construction of advanced laser intelligent manufacturing equipment” from Shanghai Lingang Area Development Administration.

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

1. Posada, J.; Toro, C.; Barandiaran, I.; Oyarzun, D.; Stricker, D.; de Amicis, R.; Pinto, E.B.; Eisert, P.; Döllner, J.; Vallarino, I. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Comput. Gr. Appl.* **2015**, *35*, 26–40.

2. Segura, Á.; Diez, H.V.; Barandiaran, I.; Arabelaz, A.; Álvarez, H.; Simões, B.; Posada, J.; García-Alonso, A.; Ugarte, R. Visual computing technologies to support the Operator 4.0. *Comput. Indust. Eng.* **2018**, *139*, 105550.

3. Posada, J.; Zorrilla, M.; Dominguez, A.; Simoes, B.; Eisert, P.; Stricker, D.; Rambach, J.; Döllner, J.; Guevara, M. Graphics and media technologies for operators in industry 4.0. *IEEE Comput. Gr. Appl.* **2018**, *38*, 119–132.

4. Roy, S.; Edan, Y. Investigating joint-action in short-cycle repetitive handover tasks: The role of giver versus receiver and its implications for human-robot collaborative system design. *Int. J. Soc. Robot.* **2018**, 1, 1–16.

5. Someshwar, R.; Edan, Y. Givers & Receivers perceive handover tasks differently: Implications for Human-Robot collaborative system design. *arXiv* 2017, arXiv:1708.06207.

6. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266.

7. Someshwar, R.; Kerner, Y. Optimization of waiting time in HR coordination. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 1918–1923.

8. Michalos, G.; Makris, S.; Tsarouchi, P.; Guasch, T.; Kontovrakis, D.; Chryssoulouris, G. Design considerations for safe human-robot collaborative workplaces. *Proc. CIRP* **2015**, *37*, 248–253.

9. Michalos, G.; Makris, S.; Spiliotopoulos, I.; Misios, I.; Tsarouchi, P.; Chryssoulouris, G. ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. *Proc. CIRP* **2014**, *23*, 71–76.

10. Someshwar, R.; Meyer, J.; Edan, Y. Models and methods for HR synchronization. *IFAC Proc. Vol.* **2012**, *45*, 829–834.

11. Wang, L.; Gao, R.; Váncaza, J.; Krüger, J.; Wang, X.V.; Makris, S.; Chryssoulouris, G. Symbiotic human-robot collaborative assembly. *CIRP Ann.* **2019**, *68*, 701–726.

12. Vojir, T.; Noskova, J.; Matas, J. Robust scale-adaptive mean-shift for tracking. *Pattern Recognit. Lett.* **2014**, *49*, 250–258.

13. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141.

14. Kwon, J.; Lee, K.M. Visual tracking decomposition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1269–1276.

15. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1409–1422.

16. Danelljan, M.; Shahbaz Khan, F.; Felsberg, M.; Van de Weijer, J. Adaptive color attributes for real-time visual tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1090–1097.

17. 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; Springer: Berlin, Germany, 2014; pp. 188–203.

18. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 850–865.

19. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575.

20. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016; pp. 4293–4302.
21. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017; pp. 6638–6646.

22. 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; Springer: Berlin, Germany, 2012; pp. 702–715.

23. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.

24. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern, Las Vegas, NV, USA, 27-30 June 2016; pp. 1401–1409.

25. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 37, 583–596.

26. Tian, Y.; Hu, H.; Cui, H.; Yang, S.; Qi, J.; Xu, Z.; Li, L. Three-dimensional surface microtopography recovery from a multifocus image sequence using an omnidirectional modified Laplacian operator with adaptive window size. Appl. Opt. 2017, 56, 6300–6310.

27. Danelljan, M.; Häger, 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.

28. Tian, Y.; Luo, J.; Zhang, W.; Jia, T.; Wang, A.; Li, L. Multifocus image fusion in q-shift dtcwt domain using various fusion rules. Math. Probl. Eng. 2016, 2016.

29. 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.

30. 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.

31. Wang, Y.; Wei, X.; Shen, H.; Tang, X.; Yu, H. Adaptive model updating for robust object tracking. Signal Proc. Image Commun. 2020, 80, 115656.

32. Lee, H.; Choi, S.; Kim, C. A memory model based on the siamese network for long-term tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8-14 September 2018.

33. Kristan, M.; Matas, J.; Leonardis, A.; Vojíř, T.; Pflugfelder, R.; Fernandez, G.; Nebehay, G.; Porikli, F.; Čehovin, L. A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. 2016, 38, 2137–2155.

34. Tian, Y.; Jia, Y.; Li, L.; Huang, Z.; Wang, W. Research on Modeling and Analysis of Generative Conversational System Based on Optimal Joint Structural and Linguistic Model. Sensors 2019, 19, 1675.

35. Kuai, Y.; Wen, G.; Li, D. Multi-Task Hierarchical Feature Learning for Real-Time Visual Tracking. IEEE Sensors J. 2018, 19, 1961–1968.

36. Li, D.; Wen, G.; Kuai, Y.; Porikli, F. Learning padless correlation filters for boundary-effect free tracking. IEEE Sensors J. 2018, 18, 7721–7729.

37. Babenko, B.; Yang, M.H.; Belongie, S. Visual tracking with online multiple instance learning. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 983–990.

38. Breitenstein, M.D.; Reichlin, F.; Leibe, B.; Koller-Meier, E.; Van Gool, L. Robust tracking-by-detection using a detector confidence particle filter. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision. IEEE, Kyoto, Japan, 29 September–2 October 2009; pp. 1515–1522.

39. 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.

40. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 37, 1834–1848.

41. Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Porikli, F.; Khajenezhad, A.; Salahedin, A.; Soltani-Farani, A.; Zarezade, A.; Petrosino, A.; others. The Visual Object Tracking VOT2013 challenge results.
42. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016, Springer: Berlin, Germany, 2016; pp. 749–765.

43. 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, 8–16 October 2016, Springer: Berlin, Germany, 2016; pp. 472–488.