Article

Reliable Memory Model for Visual Tracking

Daohui Ge¹ , Ruyi Liu¹, Yunan Li¹ and Qiguang Miao^{1,2,*}

¹ School of Computer Science and Technology, Xidian University, Xi'an 710071, China; dhge@stu.xidian.edu.cn (D.G.); ruyiliu@xidian.edu.cn (R.L.); yunanli@xidian.edu.cn (Y.L.)

² Xi'an Key Laboratory of Big Data and Intelligent Vision, Xidian University, Xi'an 710071, China

* Correspondence: qgmiao@mail.xidian.edu.cn

Abstract: Effectively learning the appearance change of a target is the key point of an online tracker. When occlusion and misalignment occur, the tracking results usually contain a great amount of background information, which heavily affects the ability of a tracker to distinguish between targets and backgrounds, eventually leading to tracking failure. To solve this problem, we propose a simple and robust reliable memory model. In particular, an adaptive evaluation strategy (AES) is proposed to assess the reliability of tracking results. AES combines the confidence of the tracker predictions and the similarity distance, which is between the current predicted result and the existing tracking results. Based on the reliable results of AES selection, we designed an active–frozen memory model to store reliable results. Training samples stored in active memory are used to update the tracker, while frozen memory temporarily stores inactive samples. The active–frozen memory model maintains the diversity of samples while satisfying the limitation of storage. We performed comprehensive experiments on five benchmarks: OTB-2013, OTB-2015, UAV123, Temple-color-128, and VOT2016. The experimental results show that our tracker achieves state-of-the-art performance.

Keywords: online update; reliable evaluation strategy; active–frozen memory model; visual tracking

Citation: Ge, D.; Liu, R.; Li, Y.; Miao, G. Reliable Memory Model for Visual Tracking. *Electronics* **2021**, *10*, 2488.

<https://doi.org/10.3390/electronics10202488>

Academic Editor: Christos J. Bouras

Received: 5 September 2021

Accepted: 30 September 2021

Published: 13 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual tracking is a fundamental problem of computer vision that tracks targets in subsequent frames by specifying the position and size of the target in the first frame. It has been successfully applied to robots, video surveillance, and self-driving cars. There are some challenging factors, such as deformation, in-of-plane scale variation, and illumination variation. These challenges are likely to cause significant changes to the appearance of the target. Therefore, how to effectively learn the appearance change of a target is an essential issue of visual tracking.

Recently, online learning-based trackers have achieved good performance. Online updates are often employed to learn appearance changes of targets. The tracking results are collected as online training samples every frame or at fixed intervals. There are some online update strategies that have been proposed [1–6]. For example, some strategies include selecting the most confident tracking result within the fixed interval frames to update specific networks [7]; collecting two consecutive frames [2]; storing each frame in order [3,8,9]; using a convolutional neural network to update the template [4,5]; and storing all tracking results using the Gaussian Mixture Model (GMM) [1,10].

Although the functions of these online update strategies have been validated, there are still two challenges. One challenge is that tracking results are not always reliable. When misalignment, occlusion, and out-of-view occur, the tracking results are likely to contain a great amount of background information, which is regarded as noise. Unreliable tracking results reduce the ability of a tracker to distinguish between targets and backgrounds, ultimately leading to tracking failure. Another challenge is that tracking results are not appropriately stored. The predicted tracking result in each frame [8,9] or several tracking results with higher confidence [2,7] are stored. However, in these methods, there are very

few online tracking samples and also only represent the latest appearance change of the target. This can easily cause the tracker to over-fit the current appearance of target.

To solve the above challenges, we propose a robust reliable memory model that can accurately evaluate the reliability of tracking results and efficiently store all reliable results. First, we propose an adaptive evaluation strategy (AES) to assess the reliability of tracking results. AES calculates the reliability weight based on the tracking confidence of the tracker prediction and the similarity distance, which is between the current predicted result and the existing tracking results. Reliability thresholds are adaptively calculated to enhance the generalization of AES. Only reliable tracking results were selected to construct online training samples. Based on the reliable results of the AES selection, inspired by the computer storage structure, we devised an active–frozen memory model to store all reliable tracking results. Training samples stored in active memory are used to update trackers online. The frozen memory temporarily stores some of the oldest results. The active–frozen memory model maintains the diversity of training samples by exchanging samples in two memories. Combined AES and the active–frozen memory model can effectively avoid introducing background information, while avoiding tracker over-fitting to the current target appearance.

The contributions are summarized as follows:

1. We propose an adaptive evaluation strategy (AES) for the reliability of tracking results. The AES adaptively calculates the reliability threshold r by combining the similarity distance and the confidence of the tracker prediction to reduce the introduction of background information. It ensures the quality of online training samples to avoid bad online updates.
2. We propose an active–frozen memory model to efficiently store all reliable tracking results. Samples stored in active memory are used to update the tracker. The frozen memory stores some of the oldest samples. Samples exchange between the active memory and frozen memory to ensure the diversity of samples within the active memory. The active–frozen memory model avoids tracker over-fitting to current appearance changes.
3. We evaluate our proposed tracker on five benchmark datasets: OTB-2013, OTB-2015, UAV123, Temple-Color-128, and VOT2016. Our tracker obtains a 69.4 AUC score on OTB-2015. Experimental results show that our proposed tracking algorithm achieves state-of-the-art performance.

2. Related Work

When scale variability, deformation, and rotation occur, the appearance of the target tends to change significantly. How to effectively learn the appearance change of a target is an essential issue of visual tracking. Recently, most approaches utilize the tracking results as online training samples to fine-tune the tracker to learn the appearance change of targets.

Reliability evaluation of tracking results. The reliability of online training samples is key to update the tracker. There are two main strategies for constructing online training samples. One strategy is to directly use the tracking results as an online training sample, regardless of its reliability. Some trackers [1,2,8,10] collect one training sample based on the tracking result in each frame. Other trackers [3,11,12] draw in some positive and negative samples around the predicted target location. When tracking drift occurs, the tracking results are likely to contain a great amount of background information that contaminates the online training samples.

The second strategy is to only consider the confidence of the tracking results, which is predicted by the tracker. FCNT [7] collects the most confident tracking results within the intervening frames. STCT [13] sets a confidence threshold and collects the tracking results with a confidence higher than the threshold. However, the tracking results are predicted by the tracker, which is always more confident about its own predictions. Thus, incorrect tracking results are still likely to achieve high confidence. Different from the above methods, we designed a robust adaptive evaluation strategy (AES) to assess the

reliability of the tracking results. The AES not only considers the confidence of the tracking results but also considers the similarity distance between the current predicted result and the existing tracking results.

Storage of online training samples. Existing trackers construct a fixed volume of space to store online training samples. Some trackers [2,7,13] maintain a very small space, which only store one or two samples, to reduce the amount of computation. CREST [2] stores only two samples, namely the last two frames. FCNT [7] stores only one training sample within the intervening frames. STCT [13] stores the tracking result with the confidence of the tracker prediction higher than a predefined threshold. These methods only collect a small amount of tracking results, making the tracker over-fit easily to the current training samples.

Other trackers collect large amounts of tracking results in large spaces. Some positive and negative samples are stored in each frame [3,11,12]. One sample is added in each frame [8,10]. UpdateNet [4] uses the initial frame and accumulated template to estimate the optimal template for the next frame. Meta-updater [6] integrates geometric, appearance, and discriminative cues to sequential information. In particular, ECO [1] employs the Gaussian Mixture Model (GMM) to reduce the redundancy of the training samples. When the number of samples reaches the maximum capacity, the tracker discards the oldest samples, which easily causes the tracker to over-fit to the current appearance of the target. We propose an active–frozen memory model to store all reliable tracking results. The training samples stored in the active memory are used to fine-tune the tracker. The frozen memory temporarily stores the sample, whose weight is less than a threshold, as discarded the by active memory. The samples in the active memory and frozen memory are exchanged to ensure the diversity of samples in the active memory.

3. Our Approach

As mentioned earlier, the reliability of training samples is very important for the online updating of a tracker. When occlusion and tracking misalignment occur, the tracking result has a good chance to contain background information, which can be regarded as noise. When the tracker is updated with these tracking results, the ability of the tracker to distinguish between the background and the target is reduced, and eventually it can lead to poor location estimation or tracking failure. As shown in Figure 1, ECO (red box) does not consider the reliability of the tracking result and is easily affected by similar objects, scale variables, and rotation. Our approach (green box) evaluates the reliability of the result to avoid introducing noise for generating better prediction results. As we know, the reliability of tracking results is not enough of a concern for researches. We obtained two observations by analyzing the confidence of the current tracking result and the similarity distance, which is between the current predicted result and the existing tracking results. Based on the two observations, an adaptive evaluation strategy (AES) was designed to evaluate the reliability of the tracking results.

The first observation. The similarity distance between the current predicted result and the existing tracking results increases significantly when tracking drift occurs. Figure 2 shows the change of the minimum distance during the tracking process. Around the 70th frame, the target jumps, causing the appearance to significantly change and leading to the similarity distance to increase rapidly. Thus, the similarity distance can help to recognize when the tracking drift occurs.

Figure 1. Our approach is compared with the ECO [1] on three test sequences called Basketball (**top row**), CarScale (**middle row**), and Twinnings (**bottom row**). ECO (red box) does not consider the reliability of the tracking result and is easily affected by similar objects, scale variability, and rotation. Our approach (green box) evaluates the result's reliability by AES.

Figure 2. Visualization of the dynamic changes of the similarity distance on Biker. The tracking drift occurs when the target jumps around 70th frame. We can clearly observe that the similarity distance is significantly increased.

The second observation. We used the VGG network to extract semantic features and to represent the target with HOG and color name (CN) features together. The tracker has the ability to address some variations in the appearance of the target. Figure 3 shows the relationship of the similarity distance and the confidence. According to the first observation, as indicated by the purple curve, when the illumination or appearance of a target changes drastically, the similarity distance increases significantly. However, the confidence of the

current predicted result (blue curve) is still higher than the mean confidence (red curve). That is, when the appearance of the target changes significantly, the tracker can still show a high level of confidence in the current prediction results.

Figure 3. Visualization of the relationship between the confidence and similarity distance on Singer2. Even if the target's pose or appearance changes significantly (purple curve), the confidence of the current predicted result (blue curve) is still higher than the mean confidence (red curve).

The tracking results are collected as training samples to update the tracker online. Based on the reliable results of the AES selection, we designed an active-frozen memory model to maintain the diversity of results while satisfying the limitation of storage.

3.1. Adaptive Evaluation Strategy (AES) of the Reliability

Inspired by the aforementioned two observations, we propose an adaptive evaluation strategy (AES) that combines the similarity distance with the confidence of the tracker prediction to assess the reliability of tracking results.

We use $U = \{u_1, \dots, u_n\} \in R^{m \times n}$ to represent the features of tracking results and $C = \{c_1, \dots, c_n\} \in R^{1 \times n}$ to represent the confidence of the tracker prediction. For the current predicted result x , its tracking confidence is represented by t and its reliable weight is represented by V . V is composed of distance-based reliability weight V_1 and a confidence-based reliability weight V_2 . When the current predicted result x is unreliable, the V is assigned a value of zero.

We first calculated the distance-based reliability weight V_1 based on the similarity distance between the current predicted result and the existing tracking results.

$$\min_{V_1} E(V_1; r) = V_1 * \left(r - \min \sum_{i=1}^n L(x, u_i) \right) \quad s.t. \quad V_1 \in \{0, 1\} \quad (1)$$

where $L(x, y)$ calculates the Euclidean distance and r is a threshold when $L(x, y)$ is greater than r , $V_1 = 0$, and otherwise is $V_1 = 1$. The purpose of V_1 is to help the tracker to identify significant changes in the appearance of the target. The confidence-based reliability weight V_2 is calculated according to the confidence of the tracking results.

$$\min_{V_2} E(V_2) = V_2 * \left(\frac{1}{n} \sum_{i=1}^n c_i - t \right) \quad \text{s.t.} \quad V_2 \in \{0, 1\} \quad (2)$$

The tracker is robust to appearance changes of the target because of the confidence-based reliability weight V_2 . Based on the distance-based reliability weight V_1 and the confidence-based reliability weight V_2 , the reliability weight V is calculated by Equation (3).

$$V = V_1 \circ V_2 \quad (3)$$

where $\circ$ is a Hadamard product. According to Equations (1)–(3), the global optimum V^* of reliability weight V is calculated by the following:

$$V^* = \begin{cases} 1, & V_1 \circ V_2 = 1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The reliability of the tracking results can be effectively evaluated by Equation (4). The parameter r is an important threshold that determines the reliability of the current predicted result. Figure 4 shows the similarity distance between the current predicted result and the existing tracking results in different sequences. In the FleetFace sequence (yellow curve), the similarity distance is significantly smaller than the Bolt2 sequence (red curve) and BlurCar1 sequence (green curve). In the Bolt2 sequence, the similarity distance shows a significant dynamic change. The similarity distance of different sequences is remarkably different because the target has different motion states, appearance changes, and resolutions of features. According to the second observation, the confidence of the tracking result can effectively address the appearance's change of the target. We propose a method that adaptively calculates the threshold r .

Figure 4. Visualization of the similarity distance between the current predicted result and the existing tracking results in the BlurCar1 (green curve), Bolt2 (red curve), and FleetFace (yellow curve). The similarity distance of different sequences is remarkably different.

In the case of $V_1 \oplus V_2 = 1$, this indicates that the distance-based reliability weight V_1 is different from the confidence-based reliability weight V_2 . When $V_2 = 1$, this indicates that the appearance of the target has changed significantly. The threshold r should be increased to select more tracking results as online tracking samples. When $V_2 = 0$, this indicates that the tracker is not certain about its own predictions. Although the new tracking results are close enough to the existing tracking results, we believe that the threshold r should be reduced to ensure the quality of the current predicted result. The threshold r can be adaptively calculated by the following formula:

$$r = r + w * \left[r - \min \sum_{i=1}^n L(x, u_i) \right] * (V_1 \oplus V_2) \quad (5)$$

where w represents the pace for each calculation.

3.2. Active-Frozen Memory Model

In order to learn the appearance change of the target, tracking results are collected as training samples to update the tracker online. Most trackers [1,7,10] discard the oldest results when the number of samples reaches the maximum limit, which results in training samples that do not fully represent the appearance change of the target.

Based on reliable results of the AES selection and as inspired by the multi-level cache technique in computer storage, we propose an active-frozen memory model that stores all reliable tracking results. The structure of the active-frozen memory model is shown in Figure 5, and is a cascaded structure that can exchange components between two memories. Tracking results stored in the active memory are used to update the tracker online. Frozen memory is used to temporarily store some of the oldest results. In order to reduce computation load, following the [1], we used the Gaussian Mixture Model (GMM) to fuse tracking results in each memory. The two closest components, namely K and S in GMM, are merged into one, specifically component G .

$$W_G = W_K + W_S, \quad \overline{X}_G = \frac{W_K \overline{X}_K + W_S \overline{X}_S}{W_K + W_S} \quad (6)$$

Figure 5. The structure of the active-frozen memory model (top row). There are two operations (below row), namely transfer component and exchange component. Only reliable tracking results are stored and are otherwise discarded directly. The active-frozen memory model guarantees the diversity and reliability of tracking results in active memory by exchange operations and AES.

We first constructed a Gaussian component based on the weight W_x and mean features $\bar{X}$ of the current predicted result x . The reliability of x was evaluated by AES (see Section 3.1 for details). If the current predicted result x is reliable, it is stored in the active memory. Otherwise, it is discarded directly.

After the current predicted results are collected, we checked whether the component numbers in the active memory had reached the maximum limit and whether the weight of one component was less than the predefined threshold. If an existing component satisfies the above requirement, it is exchanged with the closest component from the frozen memory. If the frozen memory is empty, we place this component directly into the frozen memory. The active–frozen memory model guarantees the diversity and reliability of tracking results in the active memory. The stored procedure of the active–frozen memory model is illustrated in Algorithm 1.

Algorithm 1 Stored procedure of the active–frozen memory model.

Require: current predicted result x .

Ensure: active–frozen memory.

- 1: Construct a component based on the weight W_x and mean $\bar{x}$ of the new sample
 - 2: Calculate the reliability weight V of the tracking result x by AES
 - 3: **if** $V = 1$ (the tracking result x is reliable) **then**
 - 4: The tracking result x is stored in the active memory by Equation (6)
 - 5: **else**
 - 6: Discard the tracking result X directly
 - 7: **end if**
 - 8: **if** the number of components in the active memory reaches the maximum limit and one component with the weight is less than the threshold **then**
 - 9: **if** the frozen memory is empty **then**
 - 10: Put the component into the frozen memory directly
 - 11: **else**
 - 12: Exchange with the closest component from the frozen memory
 - 13: **end if**
 - 14: **end if**
 - 15: **return** active–frozen memory.
-

3.3. Model Update

In recent trackers [1,7,12,14], a sparse update scheme was employed. The tracker, which takes collected tracking results as online training samples, is updated every N_s frames and each update performs a fixed number N_i of iteration optimization algorithms. The sparse update scheme not only reduces the computations but also reduces the over-fitting to the recent online training samples.

We also utilized the sparse update scheme in our approach. Only the training samples stored in the active memory were used to update our tracker (see Section 3.2 for details). When the current predicted result was unreliable, the active memory did not change because the predicted result was discarded directly. Thus, before updating the tracker, we detected whether the active memory changes in the N_s frame, that is, whether there were new tracking results to be collected. If the active memory had not changed, indicating the N_s tracking results were unreliable, we reduced the number of iterations N_i of the optimization algorithms to avoid the tracker over-fitting to existing online training samples. Otherwise, we performed N_i times of iteration optimization algorithms.

4. Experiments

We validated the performance of our tracker on five benchmark datasets, including OTB-2013 [15], OTB-2015 [16], UVA123 [17], Temple-color-128 [18], and VOT2016 [19].

4.1. Implementation Details

Our tracker was implemented in Pytorch. We initialized our tracker using the method proposed in [1]. The VGG-m network was used as a feature extractor to capture the *Conv1* (the first convolutional layer) and *Conv5* (the last convolutional layer) features, and the HOG and Color Name (CN) features were combined to represent the target. For the adaptive evaluation strategy (AES) of the reliability, the threshold r was initialized to 0. In order to obtain a reasonable value of r , the tracking results of the first 50 frames were used to adaptively calculate the value of r by Equation (5). In fact, the initial value of r had no effect on the performance of the tracker. In the first 50 frames, the pace for each calculation w was set to 0.5. In the subsequent frames, the pace w was calculated by the following formula.

$$w = \begin{cases} 0.4 * \max(c_i) + 0.6 * \frac{r}{distance_{min}}, & r > distance_{min} \\ 0.4 * \max(c_i) + 0.6 * \left(\frac{r}{distance_{min}} - 1 \right), & otherwise \end{cases} \quad (7)$$

where $distance_{min}$ represents the minimum similarity distance between the current predicted result and the existing online training samples.

For the active-frozen memory model, as presented in Section 3.2, the maximum limit of the number of training samples in the active memory and frozen memory was set to 50 and 10, respectively. We initialized the active memory with the tracking results of the first 50 frames of the sequence. The learning rate was set to 0.009. We updated the tracker every $N_s = 6$ frames. When tracking results were added to the active memory, we used the same iteration number $N_i = 5$ as in [1]. Conversely, the number of iterations N_i was set to 4. Note that all parameters settings were kept fixed for all the sequences in the dataset. It is important to note that the computational complexity of our proposed adaptive evaluation strategy (AES) and active-frozen memory model was $O(n)$, which is negligible and thus guarantees the real-time performance of the tracking.

4.2. Ablative Study

In this section, we analyze the contribution of both the adaptive evaluation strategy (AES) of the reliability and the active-frozen memory model to the tracker by performing experiments on the OTB-2013 dataset [15]. The OTB-2013 dataset contains 50 sequences that are all fully annotated. There are 11 attributes, such as occlusion, scale transformation, and deformation, which represent the challenge factors in visual tracking. Each sequence has at least one challenge factor. We used a precision plot and a success plot to evaluate the performance of the tracker. Precision plots calculate the Euclidean distance between the estimated location and the ground truth, and counts the percentage of frames that are less than a given threshold distance. The threshold was set to 20 pixels. The success plot quantitatively calculates the overlap ratio of the bounding box, where the overlap rate ranges from 0 to 1. The success plot counts the number of frames whose overlap rate is greater than a given threshold. The threshold was set to 0.5.

We chose ECO [1] as our baseline tracker and organized four comparison experiments by controlling variables, including standard ECO, only the adaptive evaluation strategy (ours-AES), only the active-frozen memory model (ours-AF memory), and our proposed approach (ours). Figure 6 shows the comparison experiment results on the OTB-2013 dataset. In the precision plot, the score of the baseline tracker was 93%. Compared with the baseline tracker, our active-frozen memory model achieved a 0.8% improvement and our adaptive evaluation strategy achieved a 1.6% improvement, which provided the greatest contribution. Our approach finally improved by 1.8%. In the success plot, the baseline tracker obtained an area-under-curve (AUC) score of 70.9%. Both the adaptive evaluation

strategy and the active-frozen memory model achieved a 0.4% improvement, and our approach achieved a 0.5% improvement compared with the baseline tracker.

Figure 6. Ablative experiments on the OTB-2013 dataset. The area-under-curve (AUC) score of the success plot and the score of the precision plot are represented in the legend, respectively.

We also analyzed the performance of the tracker under different challenge factors. Figure 7 only shows the results of the scale variation, illumination variation, in-plane rotation, and deformation challenge factors; we achieved an increase of 1%, 1%, 0.6%, and 2.6% respectively. In particular, our method can better learn the deformation of a target, which is our main purpose, i.e., learning the appearance change of a target.

Figure 7. Success plot on scale variation, illumination variation, in-plane rotation, and deformation. The AUC score of each challenge factor is shown in the legend.

AES guarantees the quality of online training samples to avoid introducing background information and the active-frozen memory model guarantees the diversity of

online training samples to prevent the tracker from over-fitting to the current target appearance. The experimental results in Figures 6 and 7 show that the adaptive evaluation strategy (AES) of the reliability and the active-frozen memory model are useful for improving the performance of the tracker.

Meanwhile, we conducted ablation experiments on VOT2016 [19] as shown in Table 1. Our tracker can reach 35 FPS with negligible computation introduced by AES and AF memory, satisfying the real-time requirement.

Table 1. Ablative experiments on the VOT2016.

	Baseline	Ours-AES	Ours-AF Memory	Ours
EAO	0.374	0.385	0.378	0.389
A	0.540	0.577	0.560	0.590
R	0.306	0.306	0.308	0.310
FPS	41	36	40	35

4.3. Comparisons to State-of-the-Art Trackers

In this section, we compare our approach with state-of-the-art trackers on five benchmark datasets: OTB-2013 [15], OTB-2015 [16], UVA123 [17], Temple-color-128 [18], and VOT2016 [19].

OTB-2013. We compared our approach with VITAL [3], ECO [1], MDNET [12], DAT [11], MCPF [20], CREST [2], CCOT [9], TRACA [21], BACF [22], DeepSRDCF [23], SRDCF [8], SiamFC [24], and 29 trackers from the OTB-2013 dataset. The experimental results are shown in Figure 8. In the precision plot, VITAL achieved the best performance. Our tracker obtained a precision score of 94.8%, second only to VITAL and more than the 0.4% and 1.8% of DAT and ECO, respectively. In the success plot, our method achieved the best performance between all the state-of-the-art trackers, obtaining an AUC score of 71.4%, which was more than the 0.4% and 0.5% of VITAL and ECO, respectively. Compared with ECO, although the adaptive evaluation strategy (AES) of the reliability and the active-frozen memory model had been added, the extra calculations were negligible and our trackers ran at the same speed as ECO.

Figure 8. Precision plot and success plot on the OTB-2013 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

OTB-2015. The OTB-2015 dataset is based on the OTB-2013 dataset, which adds 50 additional sequences and is still fully annotated. We compared our approach with recent state-of-the-art trackers: VITAL [3], ECO [1], MDNET [12], DAT [11], MCPF [20], CREST [2], CCOT [9], TRACA [21], BACF [22], DeepSRDCF [23], SRDCF [8], SiamFC [24], and 29 existing trackers from the OTB-2015 dataset. The experimental results are shown in Figure 9. Our approach achieved the best performance in both the precision and success plot, with a precision score of 92.3% and an AUC score of 69.4%, respectively. Our tracker

was 0.5% higher than VITAL and 1.3% higher than VITAL in the precision plot. Additionally, our tracker was 0.3% higher than ECO and 1.2% higher than VITAL.

Figure 9. Precision plot and success plot for the OTB-2015 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

UAV123. UAV123 is constructed by 123 video sequences and more than 110K frames, which contain 12 tracking attributes, captured from a low-altitude aerial perspective. We compared our approach with state-of-the-art trackers: ECO [1], MEEM [14], DSST [25], SRDCF [8], DCF [26], Struck [27], MUSTER [28], SAMF [29], and 31 trackers from the UAV123 dataset. Figure 10 shows the results over all the 123 sequences in the UAV123 dataset. Our tracker provided the best performance with a precision score of 74.9% and an AUC score of 52.8%. Additionally, our tracker achieved a substantial improvement over ECO [1], with a gain of 0.8% in the precision plot and a gain of 0.3% in the AUC.

Figure 10. Precision plot and success plot on the UAV123 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

VOT2016. The VOT2016 dataset contains 60 sequences with new annotations. We compared our approach with SiamDW [30], UpdateNet [4], SiamRPN [31], and ECO [1]. Table 2 shows the results of the VOT2016 dataset. Our tracker provided the best performance with an EAO score of 0.389.

Table 2. Comparison with state-of-the-art trackers on VOT2016.

	SiamRPN	SiamDW-RPN	ECO	UpdateNet	Ours
EAO	0.344	0.370	0.374	0.381	0.389
A	0.560	0.580	0.540	0.560	0.590
R	0.302	0.240	0.306	0.261	0.310
FPS	92	90	41	70	35

Temple-color-128. The Temple-color-128 dataset is constructed by 128 color sequences with ground truth and challenge factor annotations. As we all know, the color information of a target provides rich discriminative cues for inference. The purpose of this dataset was to study the use of color information for visual tracking. We compared our approach with MEEM [14], Struck [27], KCF [26], and other trackers from the Temple-color-128 dataset. The experimental results over all the sequences are shown in Figure 11. Our approach achieved the best performance in both the precision and success plot, with a precision score of 79.35.3% and an AUC score of 59.10%, respectively. Additionally, our tracker again achieved a substantial improvement over MEEM [14], with a gain of 8.54% in the precision plot and a gain of 9.10% in the AUC.

Figure 11. Precision and success plot on the Temple-color-128 dataset. The AUC and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

5. Conclusions

In this paper, we proposed a robust strategy for constructing online training samples to learn the changes of a target's appearance. The adaptive evaluation strategy (AES) combines the tracking confidence of the tracker prediction and similarity distance, which is between the current predicted result and the existing tracking results, to assess the reliability of the tracking results in order to ensure the quality of the online training samples. We also proposed an active-frozen memory model that can effectively store all reliable tracking results. Training samples stored in the active memory are employed to update the tracker. The diversity of the online training samples is ensured by sample exchange between two memories to prevent the tracker from over-fitting to the current appearance changes. Extensive experiments on five benchmark datasets show that our approach outperforms the performance of state-of-the-art trackers.

Author Contributions: Conceptualization, D.G., R.L. and Q.M.; methodology, D.G. and Y.L.; software, D.G. and R.L.; validation, D.G. and Q.M.; formal analysis, D.G., Y.L. and Q.M.; investigation, D.G. and R.L.; resources, Q.M.; data curation, D.G. and Y.L.; writing—original draft preparation, D.G. and Y.L.; writing—review and editing, D.G., R.L. and Y.L.; visualization, D.G.; supervision, D.G. and Q.M.; project administration, Y.L.; funding acquisition, Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research study was jointly funded by the National Key R&D Program of China under grant number 2018YFC0807500; the National Natural Science Foundations of China under grant numbers 61772396, 61772392, 61902296, and 62002271; Xi'an Key Laboratory of Big Data and Intelligent Vision under grant number 201805053ZD4CG37; the National Natural Science Foundation of Shaanxi Province under grant number 2020JQ-330, 2020JM-195; the China Postdoctoral Science Foundation under grant number 2019M663640; and Guangxi Key Laboratory of Trusted Software (number KX202061); the Fundamental Research Funds for the Central Universities under grant No.XJS210310.

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

References

1. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 3.
2. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.; Yang, M.-H. Crest: Convolutional residual learning for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2574–2583.
3. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.; Yang, M.-H. Vital: Visual tracking via adversarial learning. *arXiv* 2018, arXiv:1804.04273.
4. Zhang, L.; Gonzalez-Garcia, A.; Weijer, J.V.D.; Danelljan, M.; Khan, F.S. Learning the model update for siamese trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4010–4019.
5. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6162–6171.
6. Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; Yang, X. High-performance long-term tracking with meta-updater. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6298–6307.
7. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
8. 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, 11–18 December 2015; pp. 4310–4318.
9. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
10. Sun, C.; Wang, D.; Lu, H.; Yang, M.H. Correlation tracking via joint discrimination and reliability learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 489–497.
11. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep attentive tracking via reciprocative learning. *arXiv* 2018, arXiv:1810.03851.
12. 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, Honolulu, HI, USA, 21–26 July 2016; pp. 4293–4302.
13. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Stct: Sequentially training convolutional networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 1373–1381.
14. Zhang, J.; Ma, S.; Sclaroff, S. Meem: Robust tracking via multiple experts using entropy minimization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 188–203.
15. 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.
16. Wu, Y.; Lim, J.; Yang, M.-H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1834–1848. [[CrossRef](#)] [[PubMed](#)]
17. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking, In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
18. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [[CrossRef](#)] [[PubMed](#)]
19. Kristan, M.; Matas, J.; Leonardis, A.; Vojř, 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. [[CrossRef](#)] [[PubMed](#)]
20. Zhang, T.; Xu, C.; Yang, M.-H. Multi-task correlation particle filter for robust object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4335–4343.
21. Choi, J.; Chang, H.J.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Choi, J.Y. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 479–488.
22. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
23. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 11–18 December 2015; pp. 58–66.
24. 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, 11–14 October 2016; pp. 850–865.
25. 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.

-
26. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [[CrossRef](#)] [[PubMed](#)]
 27. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109. [[CrossRef](#)] [[PubMed](#)]
 28. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 749–758.
 29. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 8–11 September 2014; pp. 254–265.
 30. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
 31. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.