applied sciences

MDPI

Article

AR Long-Term Tracking Combining Multi-Attention and Template Updating

Mengru Guo and Qiang Chen *

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

* Correspondence: sues_chen@sues.edu.cn; Tel.: +86-13918351156

Abstract: Aiming at the problem that the augmented reality system is susceptible to complex scenes and easily leads to the failure of tracking registration, a long-term augmented reality tracking algorithm combining multi-attention and template updating is proposed. Firstly, we improved the ResNet-50 network to extract richer semantic features instead of AlexNet. Secondly, the attention-based feature fusion network effectively fuses the template and search area features through a combination of dual self-attention and cross attention. Dual self-attention effectively enhances the information in the context, whereas cross attention adaptively enhanced the features of both self-attention branches. Thirdly, the ORB feature-matching algorithm is utilized to match the template and search image features, with the template updated if more than 150 matching feature points are found. Lastly, the anchor frameless mechanism is adopted in the classification and regression network, resulting in a significant reduction in the number of parameters. The results of experiments conducted on various public datasets demonstrate the algorithm's high success rate and accuracy, as well as its robustness in complex environments.

Keywords: augmented reality; target tracking; Siamese network; attention mechanism; template updating

Citation: Guo, M.; Chen, Q. AR Long-Term Tracking Combining Multi-Attention and Template Updating. *Appl. Sci.* **2023**, *13*, 5015. https://doi.org/10.3390/app13085015

Academic Editor: Chengnian Long

Received: 6 March 2023
Revised: 6 April 2023
Accepted: 12 April 2023
Published: 17 April 2023

1. Introduction

Augmented Reality (AR) [1] can seamlessly integrate virtual information with the real world to achieve an immersive effect. It has been extensively used in industry, military, gaming, medicine, and other fields, bringing great convenience to production and life. A crucial technology of the AR system is the tracking registration technology, which determines whether the augmented reality system can accurately overlay virtual information into the real scene in real time. Target-tracking technology can realize the tracking and positioning of objects, scenes, people, and other targets, providing more accurate, stable and reliable information support for augmented reality applications. However, challenges such as occlusion, similar target interference, and motion blur often occur in long-term tracking, which can easily lead to tracking failure and result in an incorrect superposition of virtual objects (also known as the drift phenomenon). Therefore, it is of great practical significance for augmented reality technology to improve tracking accuracy and real-time performance of target-tracking technology in complex scenes.

In previous work, augmented reality target tracking is usually completed by tracking based on feature points, optical flow method, and correlation filter tracking algorithm. Among them, tracking methods based on feature points, such as SIFT [2], SURF [3], and ORB [4], to name a few, mainly extract robust feature points in images, generate descriptors, and then conduct feature matching to determine the target location. However, tracking only by this feature-matching method is easy to be affected by illumination, changes in the appearance of the target, and so on, leading to tracking failure, and the robustness is not high in practical application. The optical flow method uses the brightness variation between pixels to calculate the motion vector. This method is unstable in complex scenarios and will cause cumulative errors in long-term tracking, leading to tracking failure.

Traditional correlation filtering algorithms, such as MOSSE [5], KCF [6], DSST [7], and CSK [8], to name a few, usually adopt manual features, which have poor accuracy and robustness compared with depth features. So, they usually face challenges in maintaining accuracy in complex environments due to factors such as target occlusion, similar target interference, and scale transformation, leading to tracking failures. Due to the continuous development of deep learning technology in recent years, which provides new ideas for target-tracking technology, researchers have worked extensively on target tracking based on deep learning [9,10].

Among them, tracking-based Siamese networks [11–18] with both real time and robustness have attracted wide attention. Most trackers based on Siamese networks mostly use shallow convolutional networks (such as AlexNet [19]) for feature extraction. However, as shallow networks cannot extract deep semantic information, their feature expression ability is limited. At the same time, most well-known trackers (such as SINT [20], SiamFC [12], and SiamRPN [13]) usually calculate the similarity between template branches and search branches to achieve the function of feature fusion. However, it is easy to lose a lot of context information by using only this linear matching method, and it is easy to lead to tracking failure when encountering complex environments. In addition, most trackers (SiamDW [21], SiamRPN [13], and SiamFC [12]) do not have the operation of online template updating, which makes it more difficult for trackers to track targets with large appearance changes, occlusion, to name a few. Therefore, we choose the object-tracking method based on Siamese to carry on in-depth research and make some improvements on the basis of it.

To solve these problems, we have made some improvements on the basis of the Siamese network. Our contributions are as follows: (1) Feature extraction using the improved ResNet-50 [22] network to enhance the representation power of the feature embeddings. (2) We introduced attention mechanisms to capture the relationships between the features, leading to more accurate feature fusion. (3) The ORB feature-matching algorithm was used to match the feature of the template image and the search image. If the number of matched feature points was more than 150, the template will be updated with a threshold set to ensure reasonable updates, which would greatly improve the stability of the tracker. (4) Experiments on multiple public datasets demonstrate that our tracker achieves good accuracy and also meets the real-time requirement of an augmented reality system.

2. Related Work

Since MOSSE [5], a variety of excellent target-tracking algorithms with correlation filtering have come out one after another. However, because the shape and size of the target are constantly changing, the correlation filtering algorithm cannot cope with the tracking requirements in complex scenes. Deep learning can learn richer representations of information.

At present, the most popular method is target tracking based on the Siamese network. Bertinetto et al. proposed the SiamFC [12] tracker in 2016, which locates the target by calculating the similarity between the template and the search branch, setting off a research boom in target tracking based on Siamese networks. In 2018, Li et al. proposed SiamRPN [13], which introduced a regional proposal network based on SiamFC to improve tracking accuracy. However, the anchor frame operation introduced a large number of hyperparameters, which was not effective in the face of large deformation and scale change. DaSiamRPN [18] introduces template-based data enhancements based on SiamRPN [13] to improve model robustness and generalization ability. As the above methods all use a shallow backbone network (AlexNet) for feature extraction, the overall performance cannot be greatly improved. SiamRPN++ [17] uses a modified ResNet-50 network for feature extraction on the basis of SiamRPN and obtains good results.

In order to make the network pay more attention to useful information, the attention mechanism [23] has been widely used in the fields of natural language processing and computer vision in recent years, bringing new ideas for target tracking. Hu et al.

proposed that SENet [24] could strengthen important channel features and enhance network representation ability by adaptively learning the weight of each channel. In 2018, Wang et al. proposed a Non-Local [25] network, which can capture the global information of all positions in the input feature map, so that the network has stronger modeling ability. Woo et al. proposed CBAM [26] to improve model performance by integrating channel attention and spatial attention. Wang et al. added three attention mechanisms into the appearance branch, which enhanced the ability of feature representation, but limited the ability of network representation because only one of the branches was used. The trackers based on the attention mechanism above either act on only one of the branches or rely on correlation operations for feature fusion in the end. In this paper, inspired by Transformer [23], we design a dual self-attention and mutual attention module to merge the feature information of the two branches of the template and search branch without any correlation operation.

3. Methods

Our tracker consists of the backbone network, the feature fusion network, and the classification and regression network: (1) The backbone network adopts the modified ResNet-50 network to extract the features of the template image and search image and fuse the features of the last level of three layers to ensure the retention of rich foreground and background information and avoid the loss of shallow information that is too deep in the network. (2) The feature fusion network consists of dual self-attention and cross attention, which can enhance and fuse the extracted features so that the tracker can focus on useful information adaptively. (3) The ORB feature-matching algorithm was used to match the features of the template image and the search image. If the number of matched feature points was greater than 150, the template was updated and the threshold was set to update the template, which would make the tracker more stable. (4) The classification and regression network adopts the anchor-free method, leading to reduced complexity and easier implementation. The architecture of the framework of the algorithm is shown in Figure 1.

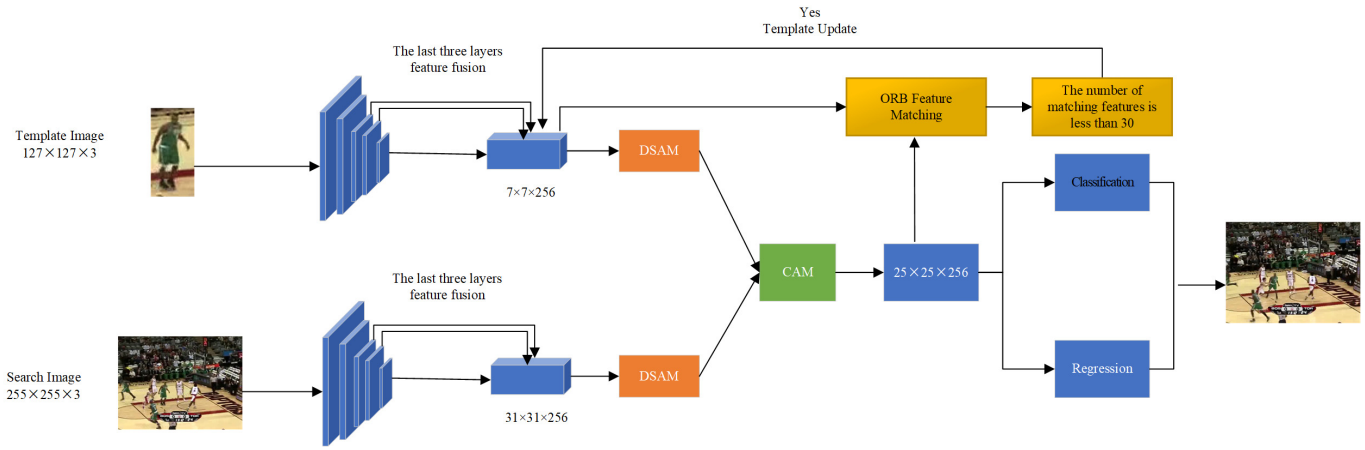

Figure 1. Architecture of our tracking framework.

3.1. Backbone

Most of the previous tracking algorithms use the AlexNet network for feature extraction, but the shallow network cannot extract rich semantic information. Therefore, we improved the ResNet-50 network instead of the shallow network to extract deeper features. Because the original ResNet network cannot put spatial information to good use and is not suitable for target tracking, the ResNet network is improved as follows:

- Reduce the effective stride of Res4 and Res5 to 8 pixels.
- In order to obtain greater feature resolution, we changed the convolution stride of the downsampling unit of the last stage of ResNet-50 to the unit convolution step.

- Adding a 1 × 1 convolution reduces the number of channels to 256, reduces the number of parameters, and keeps the same number of channels (the number is 256).
- Res3, Res4, and Res5 were fused successively, and the shallow and deep feature information was effectively fused layer by layer to obtain more refined feature information and raise the accuracy of follow-up tracking.

3.2. Feature Fusion

We proposed a feature fusion network consisting of dual self-attention (DSAM) and cross attention (CAM) by referring to the core idea of a transformer [23]. For the purpose of getting more contextual information, we use channel attention and Non-local [25] networks as inspiration to design a dual self-attention mechanism. Cross attention adaptively merges the feature maps from two branches of self-attention and establishes the relationship between the long-distant features well. The structure is shown in Figure 1.

3.2.1. Dual Self-Attention Module

The dual self-attention module consists of a channel attention network and a Non-Local network. The attention mechanism is a good reference to people's ability to observe things and improves the defect of not being able to extract rich contextual information. It can better focus on useful information, so as to facilitate the tracker to avoid the interference of chaotic background information and improve tracking accuracy. Non-Local is a network which can capture long-distance dependencies between features at any location. Channel attention mechanisms can carry out adaptive weighting for each channel to improve the effective use of relevant channel information and enhance the characterization ability of the network. Therefore, a dual self-attention based on a Non-Local network and channel attention is designed to learn more contextual information. In this work, the Non-Local attention network made use of getting more spatial information. In addition, so as to better utilize effective channel responses, we choose channel attention to calculate the channel weight coefficient to weigh channels, reduce the influence of irrelevant channels, and increase the characterization ability of networks. The structure of the DSAM module is shown in Figure 2.

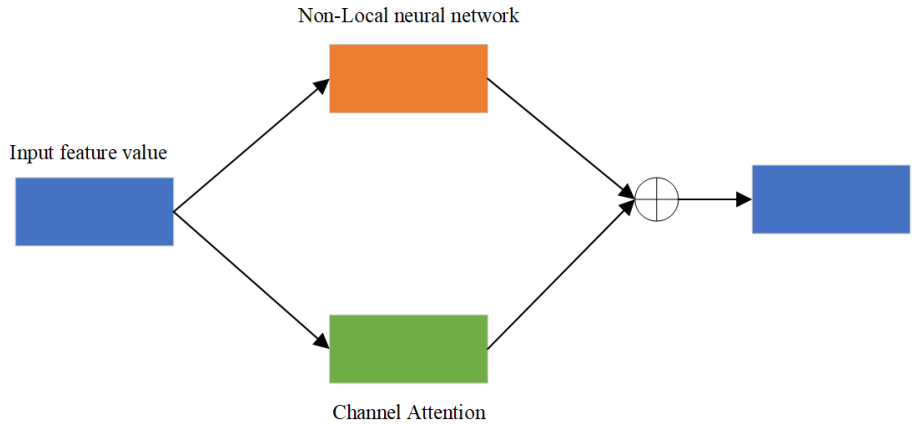

Figure 2. Dual Self-attention Module (DSAM).

The attention mechanism improves the defect that the feature extraction network cannot extract rich context information due to the limitation of the receptive field, better focuses on useful information, facilitates the tracker to avoid the interference of chaotic background information, and improves the tracking accuracy.

The channel attention mechanism is used to address the issue of irrelevant channel information interfering with model calculation and negatively impacting target positioning accuracy. By weighing each channel adaptively, the effective utilization of relevant channel information is improved, the characterization ability of the network is enhanced, and the tracking accuracy is increased. The mechanism leverages location information to identify

the richest part of the image information in each channel, thus complementing the channel's attention and helping distinguish the target from the complex background. The structure is illustrated in Figure 3.

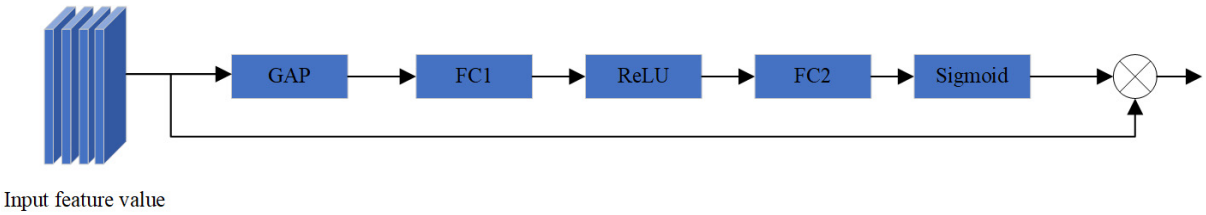

Figure 3. Channel Attention.

Non-Local attention network: different from traditional local operations such as convolution operation and cyclic operation, traditional convolution operation can only calculate the relationship between adjacent features, while Non-Local is a kind of network, which can capture the long-distance dependence relationship between any location features. Therefore, our work introduces a Non-Local network to extract location information. The structure is shown in Figure 4.

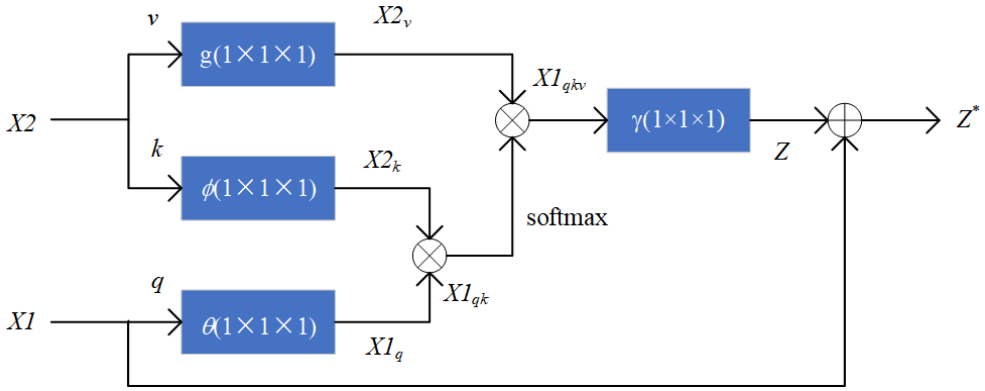

Figure 4. Non-Local Network.

The input of the Non-Local network is $X1 \in H_{X1} \times W_{X1} \times C$ and $X2 \in H_{X2} \times W_{X2} \times C$, respectively, and the input $X1$ and $X2$ are assigned to three matrices, namely, q, k, and v, respectively, and q, f, and g are the convolution operators of q, k, and v, respectively. After the 1×1 convolution of these three input matrices, the matrix transpose operation is carried out, and the matrix multiplication operation is carried out twice. After the 1×1 convolution, the addition of the original matrix $X1$ is carried out to obtain the final output Z^*. The expression is as follows:

$$X2_v = g(X2)_M, \tag{1}$$

$$X2_k = \phi(X2)_M{}^T, \tag{2}$$

$$X1_q = \theta(X1)_M, \tag{3}$$

$$X1_{qk} = softmax(X1_q \cdot X2_k), \tag{4}$$

$$X1_{qkv} = X1_{qk} \cdot X2_v, \tag{5}$$

$$Z = \gamma(X1_{qkv}), \tag{6}$$

$$Z^* = Z + X1. \tag{7}$$

After matrix multiplication of $X1_q \in H_{X1} \times W_{X1} \times C$ and $X2_k \in H_{X2} \times W_{X2} \times C$, $X1_{qk} \in (H_{X1} \times W_{X1}) \times (H_{X2} \times W_{X2})$ is obtained through softmax, and matrix multiplication with $X2_v \in H_{X2} \times W_{X2} \times C$ is to obtain $X1_{qkv}$, matrix Z is obtained through γ convolution, and the final result is obtained by adding with input $X1$ matrix.

3.2.2. Cross-Attention Module

To prevent the phenomenon of tracking drift caused by similar target interference or target occlusion, background information is particularly important to target tracking. Thus, we designed the cross attention inspired by Transformer [23], which performs feature mapping between the target template and the information of the two branches of the search area, which can retain effective background information. Additionally, to enrich the use of information from different locations, we introduced multiple attention to integrating the full range of attention. The structure is shown in Figure 5.

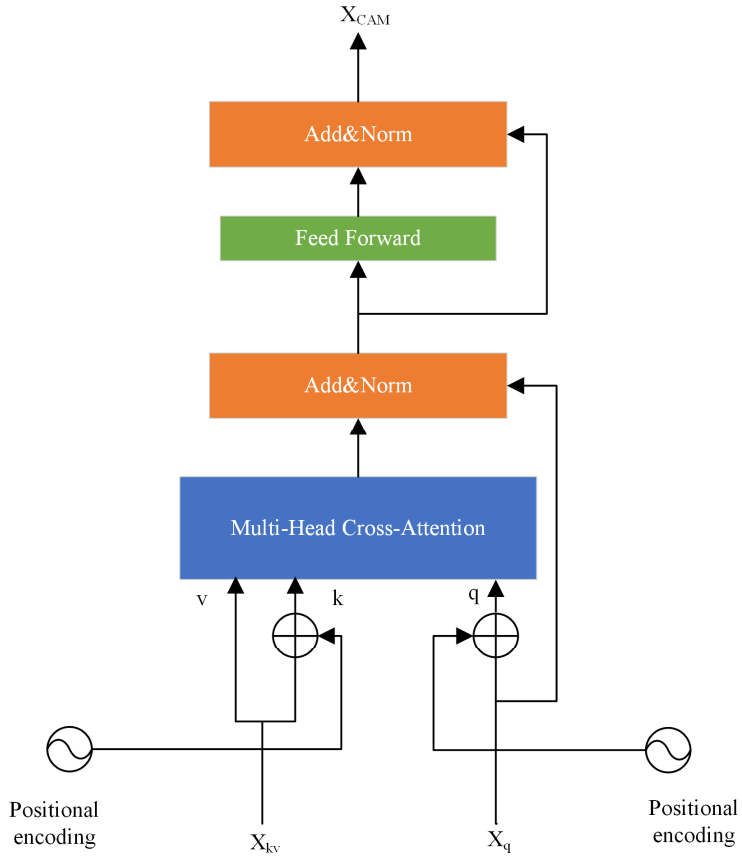

Figure 5. Cross Attention Module (CAM).

This module is composed of a residual network connecting a multi-head attention mechanism and a feedforward network, and finally, layer normalization is carried out. Among them, a feedforward network is composed of two linear transformations and a ReLU function. The calculation process of the CAM module is as follows:

$$X_{CAM} = \widetilde{X}_{CAM} + \mathrm{FFN}(\widetilde{X}_{CAM}) \tag{8}$$

$$\widetilde{X}_{CAM} = X_q + \mathrm{MultiHead}(X_q + P_q, X_{kv} + P_{kv}, X_{kv}) \tag{9}$$

where X_q is one of the input branches. Meanwhile, the spatial position code generated by the cosine function is introduced to distinguish the position information of input features. P_q is the corresponding spatial position code of X_q, X_{kv} is the input of another branch, P_{kv} is the corresponding position code of X_{kv}, X_{CAM} is the output of the CAM module. The CAM module fuses the features of the two branches and finally enhances the characterization ability of the model by means of a feedforward neural network. CAM is responsible for accepting feature mappings from two dual branches of self-attention simultaneously, forming a complete feature fusion network.

3.3. The Matching Feature Points Threshold Template Updating

Traditional Siamese network algorithms (such as SiamFC and SiamRPN, to name a few.) always set the target-tracking template to the first frame, and do not update the template during tracking. This can result in decreased tracking accuracy due to changes in the target's appearance in complex environments such as occlusion, illumination, and motion. Updating the template frequently would increase computational complexity and increase tracking errors. To balance this, this article uses thresholds to determine whether a template update is required. Firstly, the ORB feature-matching algorithm was used to match the extracted features. Then, the matched feature points threshold is set to determine whether to update the template. If the number of matched points is below the threshold (150 in this method), the target is considered lost, and the template is initialized again. If the number of matched feature points is more than the threshold, update the template.

The ORB algorithm is utilized for feature matching in this study due to its high description and matching efficiency as well as good real-time performance compared with traditional algorithms, such as SIFT and SURF. The feature-matching process is roughly as follows:

Firstly, the ORB algorithm was used to detect and describe key points in the extracted feature vectors, and the key points in the image and corresponding BRIEF descriptors were obtained. Secondly, the Hamming distance is used to match the key points. If the Hamming distance between feature points is less than the set threshold (set as 150 in this work), they are considered a match. In other words, by comparing the bit values of two feature descriptors, if different, the counter will be increased by one. The final counter result is the Hamming distance between the two binary numbers. As shown in Formula (10), ORB_{pre} and ORB_{next} are the feature descriptors of two adjacent feature points, respectively. Finally, the image is registered based on the homography matrix and the number of interior points. If the number of matching points is more than 150, the template is updated.

$$Ham(ORB_{pre}, ORB_{next}) = \sum_{0}^{255} ORB_{pre} \oplus ORB_{next} \tag{10}$$

3.4. Classification and Regression Network

The regional proposal network places the center of the anchor frame at the center of the search area with the largest responsiveness as the bounding frame of the regression. In this work, the bounding frame is directly determined by the method of classification regression for each position. This approach eliminates the need for precise adjustment of hyperparameters, which is a challenge for the tracker to deal with large deformation, our work accurately completes the classification and regression without using anchors for regional proposal, which greatly reduces the number of hyperparameters and makes the network structure simple. The structure diagram of this module is shown in Figure 6:

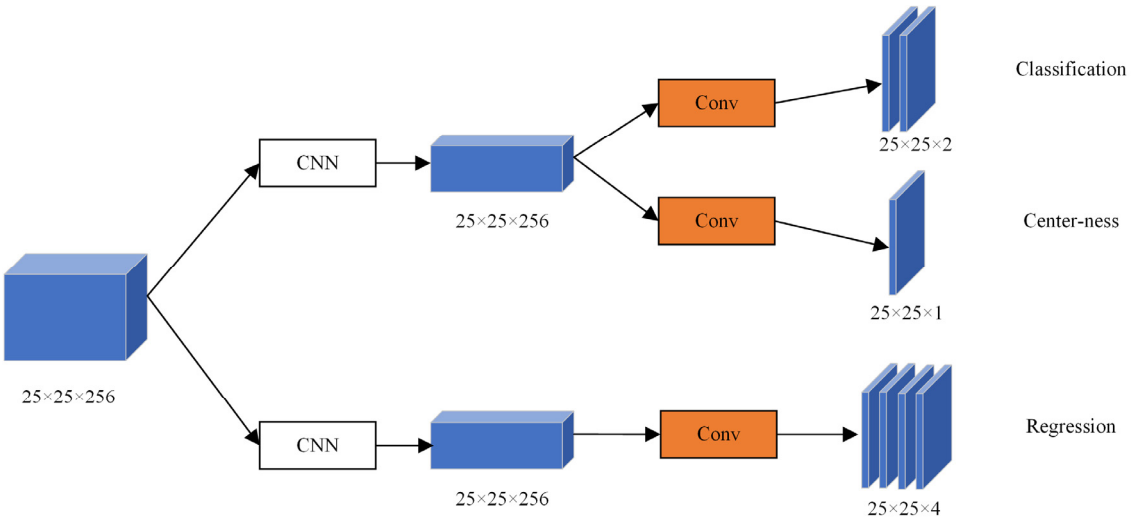

Figure 6. Classification and Regression Network.

3.5. Loss

We train many positive and negative samples to construct the loss function, which will be positive samples with the feature vector in the enclosing box, but negative samples anyway. In this work, the standard binary cross entropy loss is adopted for classification, and the definition of the formula is as follows:

$$L_c = -\sum_{pos}\left[y_{\text{pos}}\log(p_{pos}) + (1 - y_{pos})\log(1 - p_{pos})\right] \tag{11}$$

where y_{pos} is the true label, which is either 0 or 1. The location's prospects are embodied in $y_{pos} = 1$, and p_{pos} represents the probability of the prospect predicted by the model. For the regression branch, a linear combination of norm loss $\mathcal{L}_1(.,.)$ and generalized loss IoU is adopted. Regression loss can be expressed as follows:

$$L_r = \sum_{pos}\left[\alpha\mathcal{L}_{GIoU}(y_{pos}, y) + \beta\mathcal{L}_1(y_{pos}, y)\right] \tag{12}$$

A positive sample refers to an instance or data point in which the target value (y_{pos}) is equal to 1, p_{pos} is the predictive boundary box of the position, and y is the normalized true value boundary box. The regularization parameter is set to $\alpha = 1$ and $\beta = 4$.

4. Experimental Analysis

4.1. Implementation Details

The experimental platform is Ubuntu20.03 operating system, Pytorch1.7.1, Python 3.8.8, and the GPU used is NVIDIA GeForce GTX 1080Ti and 11 GB video memory.

We use COCO [27], TrackingNet [28], LaSOT [29] and GOT-10k [30] datasets to train the model. For the TrackingNet, LaSOT, and GOT-10k datasets, we sampled images from a video sequence as training samples, while for COCO data sets, we first adopted traditional data enhancement methods, such as translation and rotation, to expand the dataset.

The size of the template image is $127 \times 127 \times 3$, and the size of the search image is $255 \times 255 \times 3$, both of which are RGB three-channel graphs. After massive data training, the ResNet-50 network sets the learning rate of the trunk as le-5, the learning rate of other parameters as le-4, and the weight attenuation as le-4. We trained the network on the NVIDIA GeForce GTX 1080Ti GPU with the epoch set to 500. At the 350th epoch, the learning rate began to drop by 10 times. In the test and evaluation stage, we used the OTB100 [31], LaSOT [29], VOT-LT2020, and OxUvA datasets to evaluate the algorithm.

4.2. Analysis of Results

4.2.1. Quantitative Analysis

OTB100. The OTB100 dataset is a widely used benchmark for evaluating the performance of object-tracking algorithms in computer vision. It contains 100 video sequences of different object categories, such as pedestrians, vehicles, and animals, in various scenarios, such as in-the-wild, urban, and indoor environments. The video sequences vary in terms of object size, camera motion, object motion, and occlusions, providing a comprehensive and challenging evaluation for object-tracking algorithms. The dataset provides ground truth annotations for the objects' location, making it easier to quantify the accuracy of tracking algorithms. The OTB100 dataset has become a standard benchmark in the field of object tracking and is widely used by researchers and developers to evaluate and compare the performance of different tracking algorithms.

OTB100 dataset is a widely used benchmark for evaluating the performance of object-tracking algorithms in computer vision. It has 100 sequences, including different object categories, such as pedestrians, vehicles, and animals, in various scenarios, such as in-the-wild, urban, and indoor environments. In addition, there are hand-marks with nine attributes on the test sequences to represent challenging aspects, for example, object size, camera motion, object motion, and occlusions, providing a comprehensive and challenging evaluation of object-tracking algorithms. Two evaluation indicators are used, an accuracy score and a successful score. The accuracy score is the percentage of frames with the true distance between the center of the tracking result and the ground below 20 pixels. The success rate graph shows the percentage of successfully tracked frames at different thresholds.

We compared with other public algorithms in the OTB100 data set, and the experimental results are shown in Table 1 and Figure 7. Table 1 and Figure 7 show that our tracker achieves a success of 69%, which is 10.3% higher than SiamFC and 6.1% higher than SiamRPN.

As we can see from Table 1 and Figure 7, the success rate of our method is 0.690, and the accuracy rate is 0.889. The success rate is 15.6% higher than SiamFC and 11.4% higher than SiamRPN, and the accuracy rate is 14.8% higher than SiamFC and 7.3% higher than SiamRPN.

Several video sequences were selected to test the performance of the algorithm under different attributes such as occlusion, deformation, and motion blur, to name a few. These attributes are common challenges in the field of target tracking, which can lead to changes in the shape, position, and other features of the target or reduce the difference between the target and the background, thus affecting the tracking results. The experimental results are shown in Tables 2 and 3. The best three results are marked in red, green, and blue bold fonts, respectively. Tables 2 and 3 indicate that our tracker can achieve the top three performances under most of the attributes.

Table 1. Evaluation results on OTB100.

Trackers	Success	Precision
KCF [6]	0.477	0.696
SiamFC [12]	0.587	0.772
CFNet [32]	0.587	0.778
SiamDW [20]	0.627	0.828
SaimRPN [13]	0.629	0.847
GradNet [33]	0.639	0.861
DaSiamRPN [18]	0.658	0.880
ours	0.690	0.889

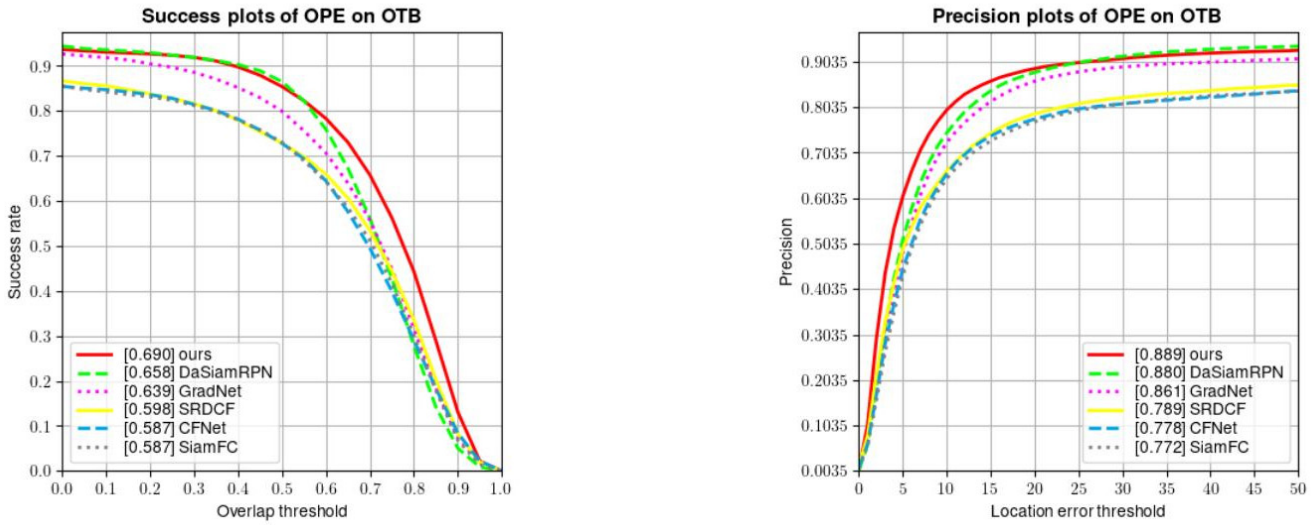

Figure 7. Performance of different algorithms on OTB100 dataset.

Table 2. Precision under different attributes. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	Illumination Variation	Occlusion	Deformation	Motion Blur	Fast Motion	Out-of-View	Background Clutters
KCF	0.728	0.749	0.740	0.650	0.602	0.650	0.753
SiamFC	0.709	0.802	0.744	0.700	0.721	0.777	0.732
SaimRPN	0.817	0.790	0.810	0.777	0.831	0.824	0.813
CFNet	0.743	0.768	0.669	0.676	0.718	0.721	0.785
Staple [34]	0.727	0.775	0.788	0.670	0.642	0.669	0.730
DaSiamRPN	0.848	0.838	0.900	0.786	0.791	0.757	0.868
SiamDW	0.841	0.855	0.906	0.716	0.732	0.734	0.847
ours	0.888	0.884	0.860	0.946	0.959	0.805	0.869

Table 3. Success under different attributes. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	Illumination Variation	Occlusion	Deformation	Motion Blur	Fast Motion	Out-of-View	Background Clutters
KCF	0.581	0.618	0.671	0.595	0.557	0.650	0.672
SiamFC	0.679	0.769	0.705	0.666	0.699	0.754	0.705
SaimRPN	0.757	0.736	0.719	0.715	0.784	0.824	0.777
CFNet	0.688	0.727	0.628	0.618	0.680	0.706	0.725
Staple	0.692	0.732	0.774	0.628	0.605	0.586	0.698
DaSiamRPN	0.822	0.817	0.877	0.731	0.758	0.761	0.847
SiamDW	0.772	0.808	0.846	0.699	0.705	0.726	0.775
ours	0.771	0.845	0.852	0.804	0.844	0.776	0.848

LaSOT. LaSOT is a large-scale dataset for object tracking. It contains a total of 1400 high-resolution sequences, of which 1120 are for training and 280 are for testing. We compared the success rate (Success) and accuracy (P,P_{Norm}) scores of various advanced trackers on LaSOT datasets. As shown in Figure 8 and Table 4, our tracker obtains a success of 65.1%, which is 13.6% higher than DaSiamRPN and 15.6% higher than SiamRPN++. At the same time, we compare the success for different attributes in LaSOT, such as ROT (the target rotates in the image), DEF (the target is deformable during tracking), IV (the illumination in the target region changes), OV (the target completely leaves the video frame), SV (the ratio of bounding box is outside the range), BC (the background has a similar appearance to the

target), and ARC (the ratio of the bounding box aspect ratio is outside the range). Table 5 indicates that our tracker performs better than others in most cases (including STARK [35]).

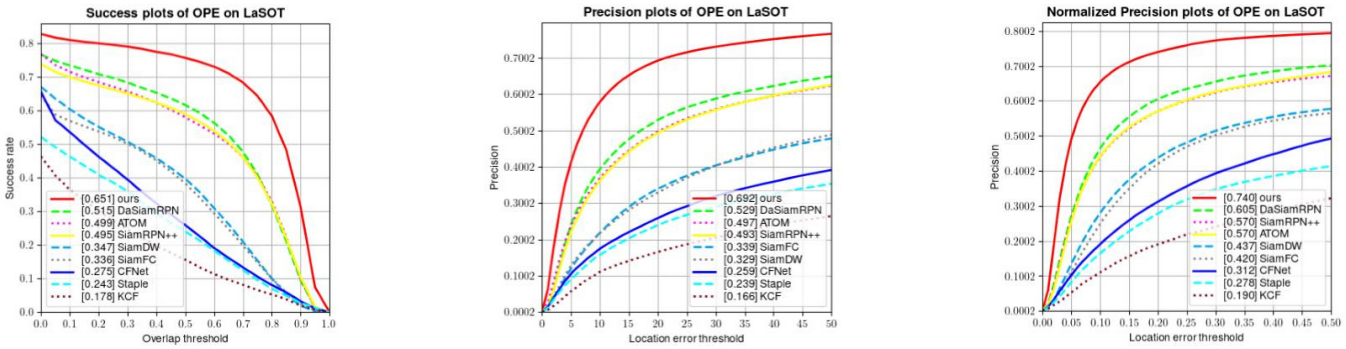

Figure 8. Performance of different algorithms on the LaSOT dataset.

Table 4. Comparison on LaSOT. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	Success	Norm Precision	Precision
KCF	0.178	0.190	0.166
Staple	0.243	0.27	0.239
SiamFC	0.336	0.420	0.339
SiamDW	0.347	0.437	0.329
SiamPPN++	0.495	0.570	0.493
ATOM	0.499	0.570	0.497
DaSiamRPN	0.515	0.605	0.529
STARK	0.671	0.770	0.713
ours	0.651	0.740	0.692

Table 5. Comparison of success rates for different attributes. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	ROT	DEF	IV	OV	SV	BC	ARC
KCF	0.198	0.256	0.179	0.345	0.278	0.197	0.252
Staple	0.256	0.287	0.367	0.268	0.398	0.376	0.296
SiamFC	0.348	0.395	0.486	0.269	0.365	0.285	0.342
SiamDW	0.375	0.435	0.414	0.376	0.459	0.398	0.452
SiamPPN++	0.469	0.548	0.534	0.598	0.497	0.398	0.586
ATOM [36]	0.496	0.534	0.601	0.576	0.612	0.614	0.409
DaSiamRPN	0.565	0.676	0.698	0.549	0.538	0.659	0.711
STARK [35]	0.790	0.765	0.734	0.854	0.789	0.613	0.654
ours	0.658	0.787	0.790	0.804	0.817	0.657	0.531

Results evaluated on the VOT-LT2020 and OxUvA:

VOT-LT2020. The VOT-LT2020 dataset is a long-term tracking dataset consisting of 50 video sequences with a total of more than 23,000 frames of images and covers various challenges in target tracking, such as target occlusion, target disappearing for long periods of time, background interference, etc. It uses Precision (Pr) and Recall (Re) to measure the performance of the tracker, and the calculation process of F-score is $F = \frac{2PrRe}{Pr+Re}$. We compare the performance of the trackers on the VOT-LT2020 and report the results in Table 6. Our tracker achieves the top three performances. Our tracker obtains a recall rate of 0.713, which is 1.8% higher than STARK-ST50.

Table 6. Comparisons on VOT-LT2020 benchmark. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	F-Score	Pr	Re
ADiMPLT	0.501	0.489	0.514
SuperDiMP	0.503	0.510	0.496
CoCoLoT	0.584	0.591	0.577
SaimDW_LT	0.601	0.618	0.585
SpLT [37]	0.635	0.587	0.544
CLGS	0.640	0.689	0.598
LTMU_B [38]	0.650	0.665	0.635
STARK-ST50	0.702	0.710	0.695
ours	0.695	0.678	0.713

OxUVA. The OxUVA dataset contained 166 test sequences and 200 development sequences using the true positive rate (TPR), true negative rate (TNR), and the maximum geometric mean of TPR and TNR (MaxGM). True negative rate (TNR) and the maximum geometric mean of TPR and TNR (MaxGM). Table 7 indicates that our tracker performs well and is second only to STARK, but STARK runs lower than us.

Table 7. Comparisons on the Ox-UvA [33] long-term dataset. The best three results are marked in red, green, and blue bold fonts, respectively.

Trackers	MaxGM	TPR	TNR
Staple	0.261	0.273	-
BACF [39]	0.281	0.316	-
EBT [40]	0.283	0.321	-
SiamFC	0.313	0.391	-
ECO-HC	0.314	0.395	-
LCT [41]	0.396	0.292	0.537
DaSiam_LT	0.415	0.689	-
TLD [42]	0.431	0.208	0.895
MBMD [43]	0.544	0.609	0.485
GlobalTrack [44]	0.603	0.574	0.633
SPLT [37]	0.622	0.498	0.776
STARK	0.782	0.841	0.727
ours	0.733	0.723	0.743

4.2.2. Visualization of the Tracking Effect

We selected some challenging video sequences on the OTB100 dataset to test our method in complex environments, and the tracking effect is shown in Figure 9.

We selected the Diving, David, Biker, Jump, and Lemming sequences in the OTB100 dataset to test our method, respectively. During the Diving video sequence, the diver's target constantly underwent scale changes, rotation, and deformation. In the David video sequence, as the target kept moving, we could find that the illumination of the environment was constantly changing, while the target also underwent scale changes, deformation, and rotation. In the Biker sequence, with the rapid rotation of the bicycle, the target goes through continuous scale change, occlusion, motion blur, fast movement, out-of-vision, and low resolution. In the Girl video sequence, the target is persistently affected by scale change, occlusion, and rotation, while in the Jump video sequence, the target has constantly undergone scale change, occlusion, deformation, motion blur, and rotation. Lemming video sequence targets constantly undergo scale change, occlusion, rapid movement, and rotation. The information displayed in Figure 9 shows that the robustness of our method remains good even when confronted with many challenges.

Figure 9. Tracking effects of the algorithm on different video sequences.

4.2.3. Tracking Registration Experiment

In addition, we adopted Unity3D as the development engine, used OpenCV and Vuforia development kits to develop the app for mobile phones in an augmented reality system, and replaced the tracking module in Vuforia with the algorithm proposed in this paper for testing. The mobile phone camera is used to collect the realistic scene image, then the tracking algorithm proposed in this paper is used to track the target, then the feature matching and pose estimation are carried out, and, finally, the virtual-real fusion is completed. We took textbooks as the target for registration and accurately registered small blocks on the textbooks. We conducted registration experiments when there is a change in scale, angle, rotation, and occlusion, respectively. The registration effects are shown in Figure 10, where (a) is the registration effect under normal conditions, (b) is the registration effect under certain occlusion, and (c) is the registration effect under different perspectives. (d) is the registration effect under rotation. As can be seen from the registration renderings in the following situations, the improved, augmented reality long-term tracking algorithm in this work can also achieve good accuracy and real-time performance in the subsequent registration stage.

4.2.4. Ablation Study

In order to prove the effectiveness of multiple attention mechanisms and template updating based on matching feature point thresholds in the proposed method, three sets of ablation experiments were conducted based on the LaSOT datasets, which were the algorithm after removing the DSAM module in our tracker and the algorithm after replacing the CAM module in the tracker with correlation operations in traditional Siamese networks, respectively. After removing the template updating module, the algorithm results are shown in Table 8. The blank said the component has been removed, ✓said this component is used.

(a) The registration effect under normal conditions

（b） The registration effect in the case of occlusion

（c） Registration effects from different perspectives

(d) The registration effect in the case of rotation

Figure 10. The registration effect of algorithm in different cases.

Table 8. Ablation results on LASOT dataset.

Method	DSAM	CAM	Template Update	Success	PNorm	P
ours	✓	✓	✓	0.651	0.740	0.692
ours	✓		✓	0.524	0.621	0.534
ours		✓	✓	0.623	0.697	0.649
ours	✓	✓		0.576	0.637	0.590

4.2.5. Speed, FLOPs, and Params

Compared with some classic trackers, a substantial decrease in the number of parameters is seen in this work, and the operation speed is greatly improved. The details are shown in Table 9. We can see that our tracker can run in real time at 47.9 *fps*. The params of our tracker are 20.579M less than STARK, and the FLOPs are 7.089G less than STARK. Our network is lighter.

Table 9. Comparison of the speed, FLOPs, and Params.

Trackers	Speed (fps)	Params (M)	FLOPs (G)
SiamFC	68.2	6.596	9.562
ATOM	55.7	18.965	50.623
SiamRPN	38.5	22.633	36.790
SaimRPN++	35.4	54.355	48.900
STARK-ST50	41.8	28.238	12.812
ours	51.9	12.659	10.723

5. Conclusions

Our method has a simple tracking framework and abandons any data post-processing and anchor frame operation and has a few parameters. Firstly, the fusion feature is used to extract the last three layers of the network to prevent the network from being too deep. Secondly, to merge the extracted features, a feature fusion network consisting of a dual self-attention module and a cross-attention module is utilized. Among them, the dual attention module adopts the combination of a Non-Local network and channel attention mechanism to enhance the contextual information and reduce computation. The cross-attention mechanism designed by Transformer is used to establish the correlation between long-distance features and effectively combine the contextual information. The tracker's stability is increased in complex surroundings. Finally, the template update module is designed to update the template in real time, avoiding the problem of tracking accuracy degradation caused by the target deformation during long-time tracking while still using the first frame as the template. A large number of experiments show that our method has shown excellent accuracy and real-time performance on multiple benchmarks, achieving a success rate of 69.0% on the OTB100 dataset, which is 15.6% higher than that of SiamFC and 11.4% higher than that of SiamRPN. A success rate of 65.1% is achieved on the LaSOT dataset. Our tracker performs well on both VOT-LT2020 and OxUvA and is second only to STARK. The number of parameters is only 12.659M, and our tracker runs faster than STARK. The network structure is simple, and the robustness shown in various challenges also indicates that our method can meet the requirement for long-term tracking in augmented reality. Future research will focus on further simplifying the network structure to better adapt to mobile augmented reality applications.

Author Contributions: M.G.; methodology, writing—original draft preparation, M.G. and Q.C.; formal analysis, investigation, resources, Q.C.; writing—review and editing, visualization, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Where data are unavailable due to privacy.

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

References

1. Thomas, P.C.; David, W.M. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In Proceedings of the Hawaii International Conference on System Sciences, Kauai, HI, USA, 7–10 January 1992; Volume 2.
2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
3. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. *Lect. Notes Comput. Sci.* **2006**, *3951*, 404–417.
4. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
5. 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.

6. 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. [CrossRef] [PubMed]
7. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, UK, 1–5 September 2014*; BMVA Press: Durham, UK, 2014.
8. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.
9. Liu, C.; Chen, X.F.; Bo, C.J.; Wang, D. Long-term Visual Tracking: Review and Experimental Comparison. *Mach. Intell. Res.* **2022**, *19*, 512–530. [CrossRef]
10. Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; Hengel, A.V.D. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1–48. [CrossRef]
11. Ondrašovič, M.; Tarábek, P. Siamese visual object tracking: A survey. *IEEE Access* **2021**, *9*, 110149–110172. [CrossRef]
12. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016*; Springer: Cham, Switzerland, 2016; pp. 850–865.
13. 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, San Juan, PR, USA, 17–19 June 1997; pp. 8971–8980.
14. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6269–6277.
15. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6728–6737.
16. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.
17. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
18. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
20. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
21. 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, 15–20 June 2019; pp. 4591–4600.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Annual Conference on Neural Information Processing Systems, Long Beach, CN, USA, 4–9 December 2017; pp. 5998–6008.
24. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
25. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014.
28. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 300–317.
29. Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Huang, M.; Liu, J.; Xu, Y.; et al. Lasot: A high-quality large-scale single object tracking benchmark. *Int. J. Comput. Vis.* **2021**, *129*, 439–461. [CrossRef]
30. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [CrossRef] [PubMed]
31. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]

32. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
33. 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, Republic of Korea, 27 October–2 November 2019; pp. 6162–6171.
34. 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 Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
35. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10448–10457.
36. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Atom: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4660–4669.
37. Yan, B.; Zhao, H.; Wang, D.; Lu, H.; Yang, X. 'Skimming-Perusal' Tracking: A framework for real-time and robust long-term tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
38. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020.
39. 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.
40. Zhu, G.; Porikli, F.; Li, H. Beyond local search: Tracking objects everywhere with instance-specific proposals. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 943–951.
41. Ma, C.; Yang, X.; Zhang, C.; Qi, J.; Lu, H. Long-term correlation tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
42. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1409–1422. [CrossRef] [PubMed]
43. Zhang, Y.; Wang, L.; Wang, D.; Qi, J.; Lu, H. Learning regression and verification networks for robust long-term tracking. *Int. J. Comput. Vis.* **2021**, *129*, 2536–2547. [CrossRef]
44. Huang, L.; Zhao, X.; Huang, K. Globaltrack: A simple and strong baseline for long-term tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11037–11044.