electronics

MDPI

Article

AMTT: An End-to-End Anchor-Based Multi-Scale Transformer Tracking Method

Yitao Zheng [1], Honggui Deng [1,*], Qiguo Xu [2] and Ni Li [1]

[1] School of Electronic Information, Central South University, Shaoshan South Road, Changsha 410012, China; 222212104@csu.edu.cn (Y.Z.); 222212065@csu.edu.cn (N.L.)
[2] School of Computer Science, Central South University, Lushan South Road, Changsha 410083, China; 234701003@csu.edu.cn
* Correspondence: denghonggui@csu.edu.cn; Tel.: +86-199-7499-4797

Abstract: Most current trackers utilize only the highest-level features to achieve faster tracking performance, making it difficult to achieve accurate tracking of small and low-resolution objects. To address this problem, we propose an end-to-end anchor-based multi-scale transformer tracking (AMTT) approach to improve the tracking performance of the network for objects of different sizes. First, we design a multi-scale feature encoder based on the deformable transformer, which better fuses the multilayer template features and search features through the self-enhancement module and cross-enhancement module to improve the attention of the whole network to objects of different sizes. Then, to reduce the computational overhead of the decoder while further enhancing the multi-scale features, we design a feature focusing block to compress the number of coded features. Finally, we introduce a feature anchor into the traditional decoder and design an anchor-based decoder, which utilizes the feature anchor to guide the decoder to adapt to changes in object scale and achieve more accurate tracking performance. To confirm the effectiveness of our proposed method, we conduct a series of experiments on different datasets such as UAV123, OTB100 and GOT10k. The results show that our adopted method exhibits highly competitive performance compared to the state-of-the-art methods in recent years.

Keywords: tracking; multi-scale; anchor; transformer; encode; decode

check for updates

Citation: Zheng, Y.; Deng, H.; Xu, Q.; Li, N. AMTT: An End-to-End Anchor-Based Multi-Scale Transformer Tracking Method. *Electronics* **2024**, *13*, 2710. https://doi.org/10.3390/electronics13142710

Academic Editors: Silvia Liberata Ullo and Li Zhang

Received: 21 June 2024
Revised: 9 July 2024
Accepted: 9 July 2024
Published: 11 July 2024

1. Introduction

The main task of visual object tracking is to obtain the feature information of the tracked target in the first frame of the video sequence so that then the tracker can automatically localize the target and calculate the position and size of the target in the subsequent frames of the video sequence based on this information. Nowadays it is used in several fields of real life, such as drone detection, live sports broadcasting, etc [1,2]. However, this field still faces numerous technical challenges, such as changing lighting conditions, occlusion of objects, and low resolution of objects. Therefore, the development of a visual object tracker that can effectively cope with these problems and has wide applicability is crucial for the further development of this technology field.

The current mainstream trackers are designed based on twin networks, which utilize a backbone network of shared weights to extract features and then fuse them for prediction [3,4]. However, most of the contemporary object tracking methods mainly rely on the last layer of high-dimensional features for feature fusion and prediction. SiamFC [5] fused the last layer of features of the twin feature extraction network through a simple cross-correlation operation to generate the final predicted response map for object tracking. But this method does not consider the problem of object size variation. Therefore, SiamRPN [6] added the RPN network to SiamFC by utilizing a predefined anchor frame to estimate the size of the object in advance. CSiam [7] increased the network depth and employed a tandem structure of multiple RPN networks to enhance the network's generalization performance and made it more sensitive to the object size. DaSiam [8] enhanced the network's immunity to interference by adjusting the dataset training strategy. However, the RPN network-based tracker had a complex structure and need to set several parameters such as the size, aspect ratio, and number of anchor frames, which not only increased the memory consumption but also affected computational efficiency. To overcome this difficulty, SiamCAR [9] proposed a novel labeling method to construct the final predicted response map into a four-layer structure, where each layer of the response map represents the distance from a point on the response map to one of the four edges of the image. By eliminating the original multi-parameter anchors set at each location in this way, the effect of hyperparameterization of the anchor settings on the tracking results is removed. TransT [10] designed a simple end-to-end tracking algorithm framework based on a transformer, which utilized the structure of codecs to achieve two-part feature fusion more efficiently while reducing the impact of hyperparameters on the results. But these methods rely heavily on a single highest dimensional feature for fusion prediction. As the backbone network goes deeper, the high-dimensional features behave as semantic information, and using only this information for tracking leads to poor model predictions for small and low-resolution targets.

In order to break through this limitation, some researchers have realized object tracking by introducing multi-layer features. SiamRPN++ [11] introduced the use of multi-layer features for fusion based on SiamRPN and reduced the number of parameters by replacing the traditional inter-correlation operation with a deep inter-correlation operation. SiamBAN [12] proposed an anchor-free multi-layer feature fusion tracker, which utilized two regression branches to directly obtain the position and dimensional size of the object, dramatically improving the speed of the tracker. However, existing multi-layer feature fusion tracking methods are all based on mutual correlation operations for fusion. This is a linear operation, making it difficult to avoid information loss during the fusion process, resulting in decreased tracking performance. Moreover, the complex structure of multi-layer correlation fusion consumes significant computational resources during actual operation, demanding more computing resources.

For the above reasons, the current visual target tracking technology has made some achievements, but the complexity of its network architecture and the lack of depth in the utilization and fusion of multi-layer features make the network's tracking ability for different size targets still require further improvement. Therefore, we propose an end-to-end Anchor-based Multi-scale Transformer Tracking (AMTT) method which aims to achieve accurate tracking of targets of different sizes through a concise network framework. One of our main innovations is the design of the multi-scale feature fusion network, which consists of three parts: a deformable attention-based multi-scale feature encoder, a feature focusing block, and an anchor-based decoder. First, inspired by Deformable DETR [13], we design a multi-scale feature encoder with a simple and stackable structure, which enables the network to efficiently fuse multiple feature maps of the template image and the search image. Then, to reduce the computational overhead of the decoder, we design a feature focusing block which is able to reduce the number of features by feature aggregating the multi-layer template features and search features passed from the encoder. Finally, we introduce anchor information in the conventional decoder. Unlike anchors in RPN networks, our anchors are a set of fixed anchors that are automatically generated in advance without setting parameters. To obtain the feature anchor, we fuse the initial anchor with the feature of the search image, and then use these feature anchor to guide the decoder to more accurately decode the location of the target from the search feature map. After validating our AMTT tracker on multiple datasets of GOT-10K [14], TrackingNet [15], OTB100 [16] and UAV123 [17], the results of the study show that our method exhibits excellent performance on every dataset, further confirming the utility of our method.

In this study, our main work includes the following four points:

1. We design a multi-scale feature encoder. This encoder is able to utilize the multi-layer features of the backbone network for fusion, which enhance the ability of the tracker to sense targets of different sizes.
2. We design a feature focusing block module inserted between the encoder and the decoder. This module can perform feature aggregation on the fused multi-layer features to achieve feature enhancement while reducing the number of feature token numbers.
3. We introduce an anchor in a traditional decoder to design an anchor-based decoder. The query of the traditional encoder is split into content query and location query, where the content query is a search feature and the location query is a predefined anchor. The combination of the two parts yields the feature anchor to guide the decoder's work and enable the tracker to predict the location and size of the target more accurately.
4. We evaluate our method on multiple datasets and compare it with state-of-the-art object tracking methods. The results validate the effectiveness of our method. In addition, we perform ablation experiments on each module to verify the independent validity of each module.

2. Related Work

2.1. Transformer in Vision

Transformer [18] was proposed in 2016 as a new architecture for machine translation, which has significant global feature fusion capability. Recently, it has been applied to the field of machine vision and is able to fuse a large range of features more efficiently than with convolutional neural networks (CNNs) [19,20]. DETR [21] is an end-to-end object detection method designed using transformer which employs CNN as a feature extraction network, utilizes transformer's codec structure for feature fusion, and performs classification and regression with a simple prediction header. This approach removes the extensive post-processing in object detection and greatly simplifies the object detection process. Deformable DETR introduces a multi-layer feature representation based on DETR and redesigns the codec module through the deformable attention to achieve the fusion of multi-layer features. This improvement solves the problem of DETR for small object detection. Conditional DETR [22] decouples the query from the decoder and introduces anchor information to accelerate the learning of the decoder, enabling the detector to be able to acquire the target location in a fine-grained manner. In this study, we are inspired by Deformable DETR and Conditional DETR, applying them to the field of object tracking. We design an encoder capable of fusing multilevel features and an anchor-based decoder that enables the network to sense targets of different sizes and target them more precisely.

2.2. Visual Object Tracking

The main strategies for visual object tracking are mainly categorized into two types: correlation filter-based object tracking methods and deep learning-based object tracking methods. Correlation filter-based tracking strategies achieve the distinction between object and background to determine the object location by designing correlation filters and using manually extracted feature images for filter training [23–25]. However, the feature extraction process of this method is more complicated and it has been gradually replaced by the rise of convolutional neural network feature extraction methods. Recently, object tracking algorithms based on the structure of twin networks have made remarkable achievements. This type of method contains three main parts: first, it extracts features from template image and searches the image using the twin neural network; second, it designs a feature fusion module to fuse the extracted features; and lastly, it performs simple prediction and regression on the fused features to determine object location through the detection head. However, most twin network trackers such as SiamFC [5], SiamRPN [6] and TransT [10] select only the last layer of features in the backbone network for fusion prediction to achieve faster tracking performance. This results in the network's lack of accuracy in tracking small and low-resolution targets and the occurrence of lost followers at length. Each position in the high-level feature map represents the semantic information within that region in the original image, so it is ineffective for tracking small and low-resolution targets in the image, and may even fail to track them. Some methods such as SiamRPN++ [11] and SiamBAN [12] use multi-layer features for fusion prediction. However, these methods are fused by hierarchical inter-correlation operations, which is a linear fusion, and the arithmetic process may lead to information loss. Moreover, the structure of hierarchical fusion is complex and computationally intensive, which cannot achieve the desired effect of multilayer feature fusion. HiFT [26] realizes the fusion interaction of multi-layer features using a codec designed by a transformer to compensate for the loss of information due to the inter-correlation operation. However, the method underutilizes the bottom-level features and is only used to assist the top-level features for interaction, indirectly limiting the further improvement of tracker performance. Therefore, in order to utilize the multi-scale features more effectively, we design an end-to-end Anchor-based Multi-scale Transformer Tracking method, through which we solve the above problems and enhance the accuracy and efficiency of object tracking.

3. Method

In this section, we present our end-to-end anchor-based multi-scale transformer tracking method. The specific structure of our method is illustrated in Figure 1. It consists of three main components: a feature extraction network, a multi-scale feature fusion network, and a network prediction head. Among them, the multi-scale feature fusion network is composed of three main modules: the multi-scale feature encoder (MFE), the feature focusing block (FFB), and the anchor-point-based decoder (AD), which are designed by us.

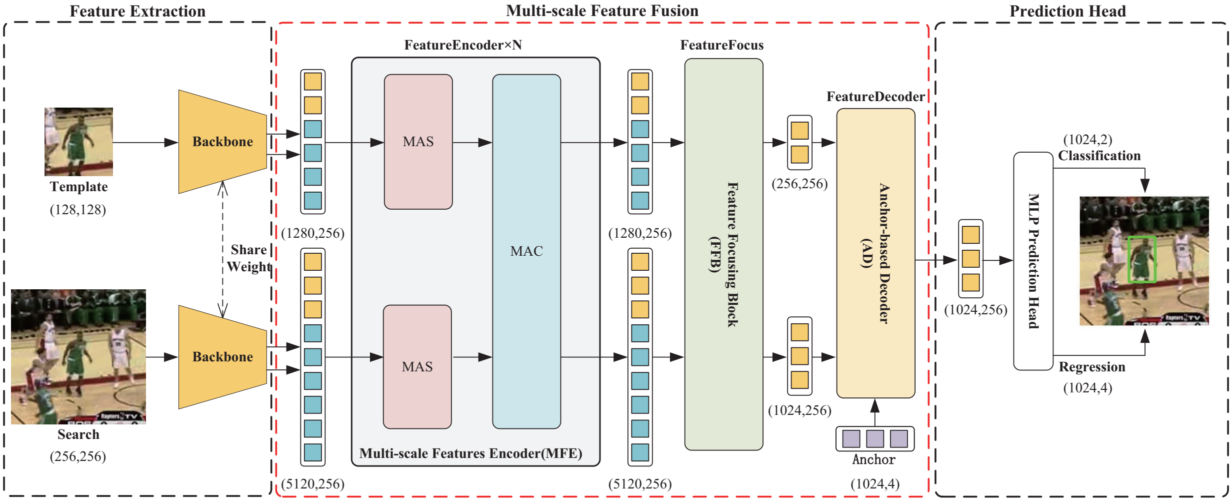

Figure 1. Overall framework of AMTT.

3.1. Feature Extraction Network

In this paper, we use the architecture of twin network, i.e., ResNet50 [27] network as a backbone to extract the features of the template image $Z \in \mathbb{R}^{3 \cdot H_{Z0} \cdot W_{Z0}}$ and the search image $X \in \mathbb{R}^{3 \cdot H_{X0} \cdot W_{X0}}$. Figure 2 illustrates the structure of the modified ResNet50. In order to make the output feature map with higher resolution, we remove the fourth block of ResNet50 and reduce the convolution step of the third block to 1 to achieve higher precision localization. We input the template image and the search image into the backbone network to obtain the multilayer features, respectively, and extract the features of the first and the third layers of them to be stacked together as the template features and the search features, which are expressed as follows in Equation (1).

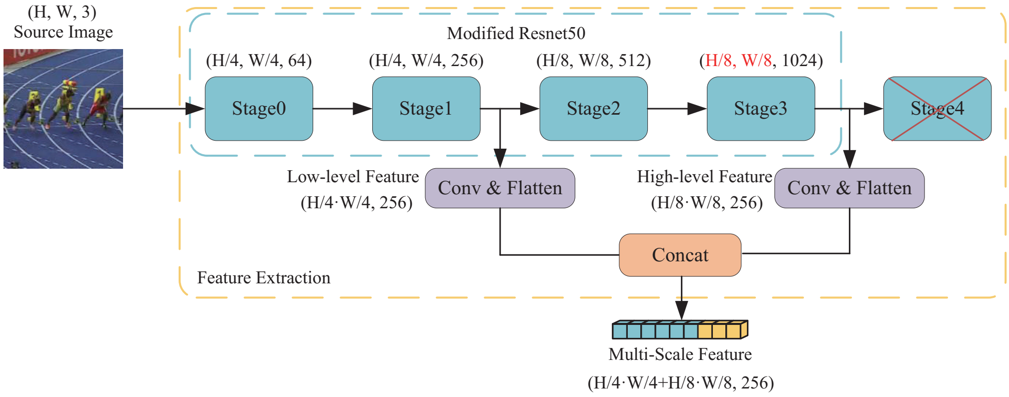

Figure 2. Structure of the modified ResNet50.

$$\Phi_k(Z) \in \mathbb{R}^{dim_k \cdot H_{Zk} \cdot W_{Zk}}, k = 1, 3, \tag{1}$$

$$\Phi_k(X) \in \mathbb{R}^{dim_k \cdot H_{Xk} \cdot W_{Xk}}, k = 1, 3, \tag{2}$$

where, $\Phi_k()$ is defined as the kth layer feature output of the backbone network. A connectivity layer is introduced to unify the feature depths of different layers to dim = 256. Finally, the width and height dimensions are spread out into one dimension and the multilevel features are stitched together in the new dimension. Thus, the final feature output of the template image and the search image after the backbone network is obtained as

$$F_Z \in \mathbb{R}^{dim_k \cdot (H_{z1} \cdot W_{z1} + H_{z3} \cdot W_{z3})} = \mathbb{R}^{dim \cdot N_z}, \tag{3}$$

$$F_X \in \mathbb{R}^{dim_k \cdot (H_{x1} \cdot W_{x1} + H_{x3} \cdot W_{x3})} = \mathbb{R}^{dim \cdot N_x}. \tag{4}$$

3.2. Multi-Scale Feature Fusion Networks

In the following section, we describe the multi-scale feature fusion network we designed, which includes three core components: a multi-scale feature encoder, a feature focusing block, and an anchor-based decoder.

3.2.1. Multi-Scale Feature Encoder

The encoder processes multi-scale search and template image features from the backbone network. First, attention self-enhancement techniques are utilized to integrate both bottom- and top-level features. This enhances the network's ability to attend to objects at different scales. Then, a cross-enhancement technique is used to combine the template and search image features to enhance the search features by utilizing the target markers in the template features. Multi-scale features that incorporate low-dimensional spatial details and high-dimensional semantic data are finally generated. Our proposed multi-scale feature encoder is shown in Figure 3, which is obtained by superposition of multiple encoder units, where the encoder units are obtained by connecting the multi-scale attention self-enhancement (MAS) module and the multi-scale attention cross-enhancement (MAC) module in series. The structure and design ideas of MAS and MAC are described below.

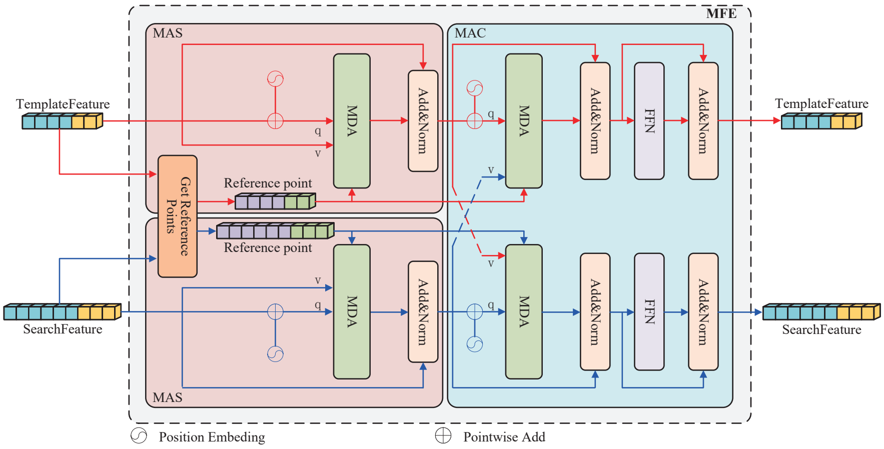

Figure 3. Structure of the multi-scale feature encoder.

Multi-scale Attention Self-enhancement Module (MAS): Its structure is shown in the pink box to the left of Figure 3. We introduce the multi-scale deformable attention (MDA) mechanism for feature enhancement. Considering that there is no position information in the multi-level feature output of backbone network, we generate multi-level position embedding (mPE) by combining sinusoidal position embedding and hierarchical embedding. It is added with multi-level features to obtain MDA query, and the multi-level features are separately introduced as the value of MDA. Finally, the reference points of each position on the feature map are directly generated by the function. The output of the MDA is summed with the value and passed through a LayerNorm layer with a dimensional depth of 256 to obtain the output of the MAS. The search image as an example with the equation is expressed as follows:

$$Q_X = Add(F_X, mPE_X), \tag{5}$$

$$V_X = F_X, \tag{6}$$

$$MAS_{out}(X) = Norm(MDA(Q_X, V_X, RP_X) + V_X), \tag{7}$$

where Q_X stands for the query of MDA, F_X stands for the search feature, mPE_X stands for the search location, V_X stands for the value of MDA, RP_X stands for the reference point with the search feature as the datum and MAS_{out} is the output of MAS. The computation of multi-scale deformable attention is as follows:

$$MDA(x_q, X, P_q) = \sum_{m=1}^{M} W_m [\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} W_m^{'} X^l (\Phi_l(\tilde{P}_q) + \Delta P_{mlqk})], \tag{8}$$

where $x_q \in \mathbb{R}^{dim \cdot 1}$ denotes a query vector in the vector space of $X \in \mathbb{R}^{dim \cdot N_x}$, P_q denotes the relative position of the query on the whole feature map, W_m represents the multi-head weight matrix, A_{mlqk} represents the relative position $\overline{P}_q$ of x_q under the lth scale feature map in the mth head versus the attentional weight of the kth position $X^l(\phi_l(\overline{P}_q) + \Delta P_{mlqk})$ in the scale and $W_m^{'}$ is the Value transformation matrix, satisfying

$$\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} = 1, \tag{9}$$

It can be seen from Equation (8) that x_q is not weighted with all vectors in the X space in the deformable attention operation, but the weighting operation is carried out at a specific location, where M represents the number of multiple heads of multi-head attention, L represents the number of feature map scales and K represents the number of reference points. With this deformable attention, the amount of computation can be greatly reduced without degrading the information interaction, allowing for each query to focus only on the information around itself that is relevant to it. Figure 4 illustrates the computation process of deformable attention through graphical representation.

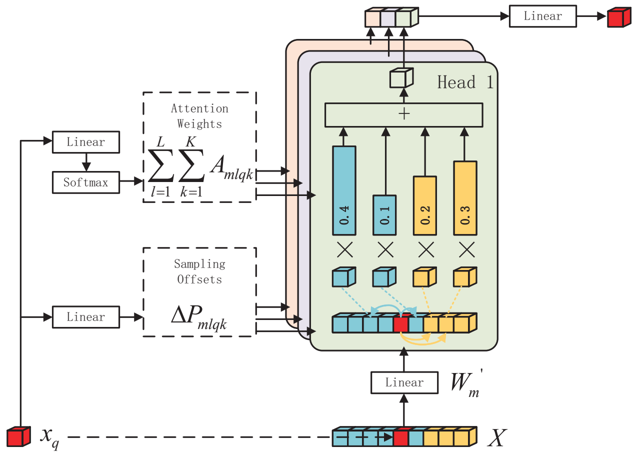

Figure 4. Illustration of a multi-scale deformable attention mechanism.

Multi-scale Attention Cross-enhance-ment Module (MAC): Its structure is shown in the blue box to the right of Figure 3. We take the MAS output of the search branch as the input to this module, and the reference coordinates are passed in by the MAS module. Unlike MAS, the Value of a MAC consists of the MAS output in the template branch. By simply adjusting the input composition of MDA, the network structure in the blue box can obtain the ability of cross-fusion. Then, the fusing features of MDA are sent to the FFN module to further enhance the MAC fitting capability. The FFN module is composed of a Relu layer and two fully connected layers, represented by the following Equation (10):

$$FFN(X) = ReLU(x \cdot W_1 + b_1) \cdot W_2 + b_2, \tag{10}$$

where W_1, W_2 are the weight matrices of the two linear layers, respectively, and the subscript is the index of each layer. Thus, the multi-scale attention cross-enhancement module can be generalized as

$$Q_X = Add(MAS_{out}(X), mPE_X), \tag{11}$$

$$V_X = MAS_{out}(Z), \tag{12}$$

$$Res_{out} = Norm(MDA(Q_X, V_X, RP_X) + V_X), \tag{13}$$

$$MAC_{out}(X) = Norm(FFN(Res_{out}) + Res_{out}), \tag{14}$$

where Q_X is the query of MDA, $MAS_{out}(X)$ is the MAS output result of the previous level search image, mPE_X is the search position, $MAS_{out}(Z)$ is the MAS output result of the template image, V_X is the value of MDA, RP_X is the reference point based on the search feature, Res_{out} is the output of MDA after passing the $Add\&Norm$ intermediate value and MAC_{out} is the final output value of MAC.

3.2.2. Feature Focusing Block

To reduce computational load and improve encoding performance, we introduce a feature focusing block. This module extracts and enhances high-dimensional features from the multi-scale features, significantly reducing the number of decoded features without losing critical information. Figure 5 illustrates the exact construction of the feature focusing module. Similar to the multi-scale feature encoder, we decided to use MDA as the attention mechanism for this module to reduce the overhead of multiscale feature computation. First, we extract the high-dimensional features from the multi-scale features sent by the

MFE. At the same time, we extract location code P_H from the location code that matches the high-dimensional features. Next, we use the spatial feature enhancement module, as shown in the blue box in Figure 5, to enhance the information-rich vectors in the high-dimensional features.

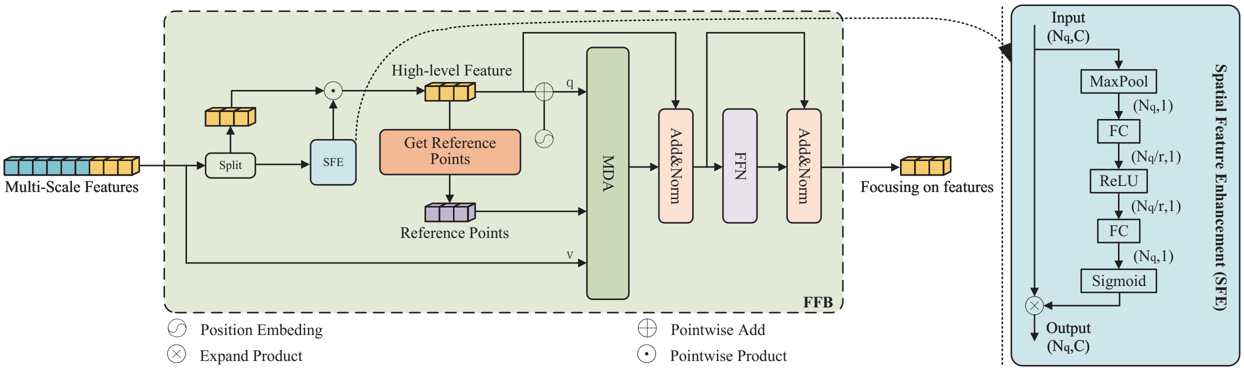

Figure 5. Structure of the feature focusing block.

In order to avoid gradient diffusion in training, we use the spatial feature enhancement (SFE) module as a side path and dot-multiply it with higher-dimensional features to obtain higher-dimensional features X_H. X_H is combined with P_H to obtain the query input of MDA. X_H receives the Reference point input through the Get Reference Points module, and then the multi-scale features of MFE output are taken as the Value input of MDA. Finally, the output of MDA is residually linked with the high-level features and input to FFN to obtain the final focusing on features.

3.2.3. Anchor-Based Decoder

In the decoding stage, we introduce anchor information in the traditional encoder to design an anchor-based decoder. The tracking results obtained from ordinary position information-assisted query decoding are not satisfactory [22]. Therefore, we we consider decoding query into the content query and the position query, where the position query utilizes the feature anchor instead, and the decoder is able to decode finer tracking results through the guidance of the feature anchor. Figure 6 illustrates the construction of the anchor-based decoder. In the decoding process, we use the traditional multi-head attention (MHA) mechanism [18]. In the previous stage, the feature focusing block greatly reduces the number of feature vectors, which allows for the multi-head self-attention algorithm to effectively model the image at the most economical cost.

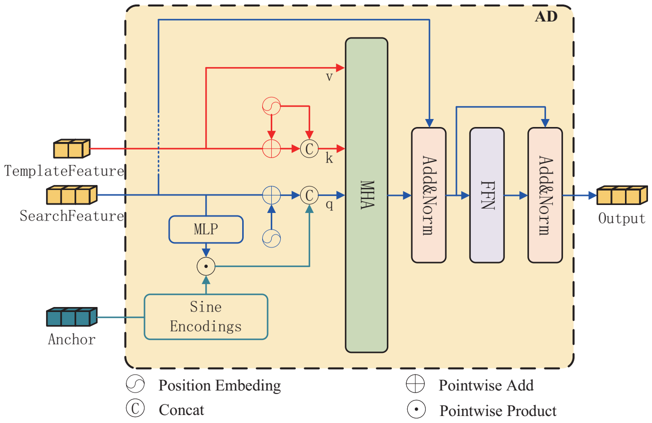

Figure 6. Structure of the anchor-based decoder.

The init anchor is a four-dimensional bounding box$\in [1024, 4]$ pre-generated at each location of the search feature map. We extend the second dimension to 256 using sinusoidal coding to facilitate interaction with the search features. Search features align the channel dimension with the anchor through a layer of ML and dot-multiply the two to obtain a feature anchor with search features, as shown in Figure 7.

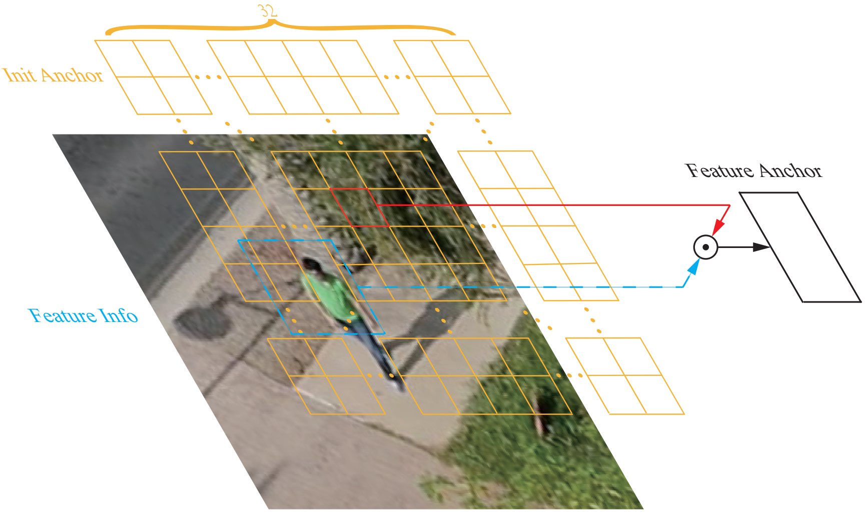

Figure 7. Illustration of the feature anchor frame generation mechanism.

The feature anchor is used as position query, and the search features and position embedding are summed to obtain a content query. The two parts of the query are spliced together in the channel direction to obtain the final multi-attention query. The template features serve as the value of multi-head attention. To align with the query dimension, the template features are added with the location code and then spliced with the position embedding to obtain the final input as the Key. The multi-head attention module is used to decode the search features by using the template features, and the feature box is used to refine the tracking results to find out the position with the highest matching degree between the search features and the template features.

3.3. Prediction Head and Loss Function

After being processed by the multi-scale feature fusion network, the output feature maps not only contain rich semantic information, but also have spatial information, while interferences and background information are effectively suppressed in the feature maps. Therefore, after the decoding is completed, the final classification correspondence map and regression response map are generated by simple MLP classification branching as well as MLP regression branching.

We realize the labeling marking all points corresponding to the target box on the response graph as positive samples and other places as negative samples. At the same time, in order to reduce the large gap between the number of positive samples compared to the number of negative samples due to the small area corresponding to the target true box on the response map, we reduce the loss of negative samples by 16 times for balancing the training of the whole network. We utilize the cross-entropy loss function for binary classification as the classification shoots the function for positive and negative samples, defined as follows:

$$\mathcal{L}_{cls} = -\sum_{j}[y_i \cdot log(p_j) + (1 - y_j) \cdot log(1 - p_j)]. \tag{15}$$

where y_i represents the true value, $y_i = 1$ is for the foreground and p_j is the probability value of the foreground in the network prediction result. We use two metrics, the L1 loss and the generalized IoU loss GIoU [28], and combine them to arrive at the final regression loss, defined as follows:

$$\mathcal{L}_{reg} = \sum_j [\mathbb{1}_{\{y_i=1\}} \lambda_G \cdot \mathcal{L}_{GIoU}(b_j, \hat{b}) + \lambda_1 \cdot \mathcal{L}_1(b_j, \hat{b})], \tag{16}$$

In calculating the regression loss, we only use the predicted data of the positive sample, where b_j represents the predicted result of the jth bounding box and $\hat{b}$ represents the actual bounding box after normalization. Both λ_G and λ_1 are parameter settings for regularization, which we choose to be 2 and 5, respectively.

4. Experiments and Results

4.1. Experimental Details

Our experiment is divided into two parts, the training phase and the inference phase, and uses a total of six public datasets including GOT-10k [14], TrackingNet [15], LaSOT [29], COCO [30], OTB100 [16] and UAV123 [17]. The connection between the datasets and the model is shown in Figure 8.

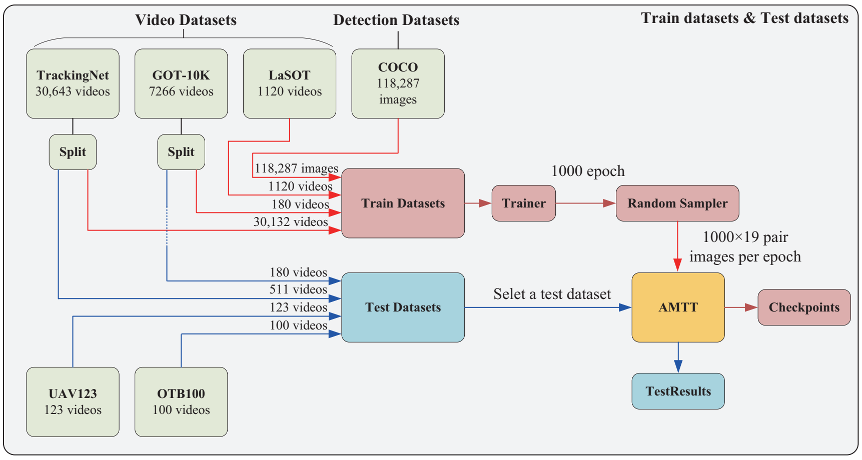

Figure 8. The figure illustrates the structure and data volume of the entire dataset, where the red line represents the training process and the blue line represents the testing process.

Training stage: Our study employs an Intel i7-13700K processor with 32GB of RAM and a single RTX3090 GPU with 24GB of video memory. We use Ubuntu 22.04.1 with the Pytorch 1.12 framework to build our code. Four datasets, the GOT10k segmented training set, LaSOT, COCO and TrackingNet are used to train the model during the training process. Commonly used data enhancement methods such as luminance dithering and level flipping are incorporated in the training process. We directly use the image pairs extracted within 10 frames from tracking datasets GOT10k, LaSOT and TrackingNet as the training data, and for the COCO dataset, the original images are directly extracted and transformed to obtain the image pairs as the training data. The size of the template image in the training data pair is 128 × 128 and the size of the search image is 256 × 256. Our feature extraction network uses the parameter weights of ResNet50 pre-trained on ImageNet, and the parameter weights of the remaining modules are initialized using Xavier. We use $\lambda_1 = 5$, $\lambda_G = 2$ and $\lambda_{cls} = 8.334$ as the loss weighting coefficients for the L1 loss, GIoU loss, and cross-entropy loss. The model is trained using the AdamW optimizer, setting the feature extraction network and other parameter learning rates to 1×10^{-5} and 1×10^{-4}, respectively, and the weight decay to 1×10^{-4}. We set the batch size to 19 and perform 1000 iterations per epoch for a total of 1000 epochs, and reduce the learning rate to 10 times the original rate at the 500th epoch. Our main training parameters are shown in Table 1.

Table 1. The main training parameters.

Parameter	Value
Template image size	128
Search image size	256
Epoch number	1000
Batch size	19
The number of iterations per epoch	1000
Total training image pairs	19,000,000
Start learn rate	0.0001
End learn rate	0.00001
Output feature map size	32

Inference stage: A Hanning window of size 32 × 32 is first set and expanded to 1024 size. Then, we multiply it with the classification map predicted by the network to obtain the final matching map. Based on this, we select the point with the largest score on the score matching map as the location where the target is located, and then determine the coordinates and size of the predicted target on the regression response map to determine the final prediction region. The flow of a single video tracking is shown in Figure 9.

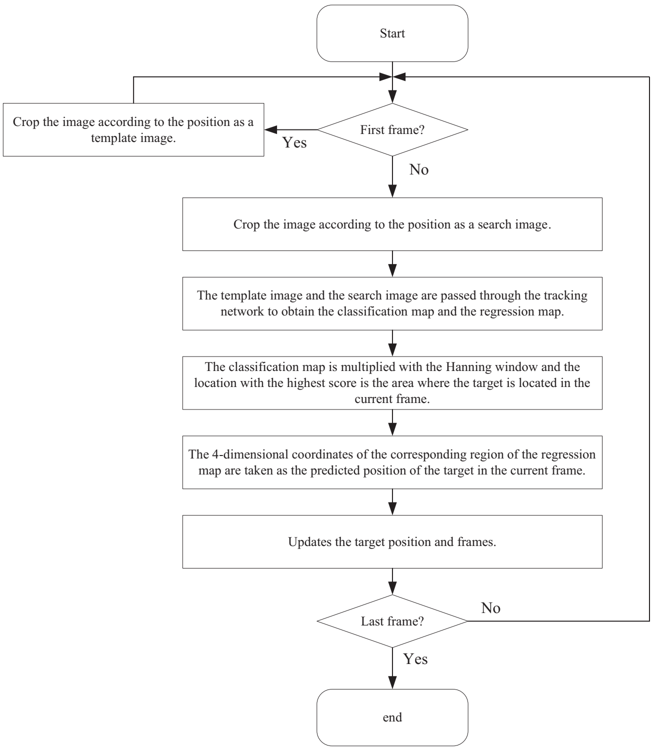
Start
First frame?
Yes
Crop the image according to the position as a template image.
No
Crop the image according to the position as a search image.
The template image and the search image are passed through the tracking network to obtain the classification map and the regression map.
The classification map is multiplied with the Hanning window and the location with the highest score is the area where the target is located in the current frame.
The 4-dimensional coordinates of the corresponding region of the regression map are taken as the predicted position of the target in the current frame.
Updates the target position and frames.
Last frame?
No
Yes
end

Figure 9. Flowchart of a single video tracking.

4.2. Comparison of Experimental Results

In this section, we analyze the proposed method in comparison with the most representative methods in the field of single-target tracking as well as the state-of-the-art methods in recent years on four different benchmarks. These benchmarks include OTB100 [16], GOT-10k [14], TrackingNet [15] and UAV123 [17].

OTB100: The OTB100 benchmark contains 100 video sequences for evaluating the parameters of Precision and Success, where the Precision plot indicates the proportion of all predicted frames whose predicted coordinate positions are within 20 pixels of the object's real coordinate positions. The Success plot indicates the proportion of all frames whose predicted frames overlap with the object's real frames by varying the overlap rate from 0 to

1, and the proportion of all frames whose predicted frames meet the requirements. It also includes eleven challenges including Illumination Variation, Occlusion, Scale Variation, Deformation and so on [16]. Table 2 shows the results on the OTB100 test set. From the table, we can see that our designed AMTT almost achieves leading performance in terms of success rate, with a 1.5% improvement compared with TransT [10] and a 1.6% improvement in terms of accuracy. We also compare the results of our designed tracker with other trackers under different challenges as shown in Figure 10. It can be seen that under the challenge of scale variation, our method achieves the best results compared with other methods thanks to the anchor-based decoder we designed, which is able to decode the size and dimensions of the target more finely with a feature anchor when the object changes. It shows that our algorithm presents better performance for low-resolution and small-scale target tracking. We visualize the tracking effect of our method with the state-of-the-art algorithm in Figure 11.

Table 2. The table shows the results on the OTB100 dataset. The best two results are shown in red and blue fonts.

	Success (%)	**Precision (%)**
AMTT (Ours)	69.6	89.9
RBO [31]	69.7	90.7
TransT [10]	68.1	88.3
Ocean [32]	66.6	89.1
ATOM [33]	66.4	87.3
DaSiamRPN [8]	65.4	87.3
SiamRPN [6]	62.6	84.2
SiamFC [5]	58.3	76.5

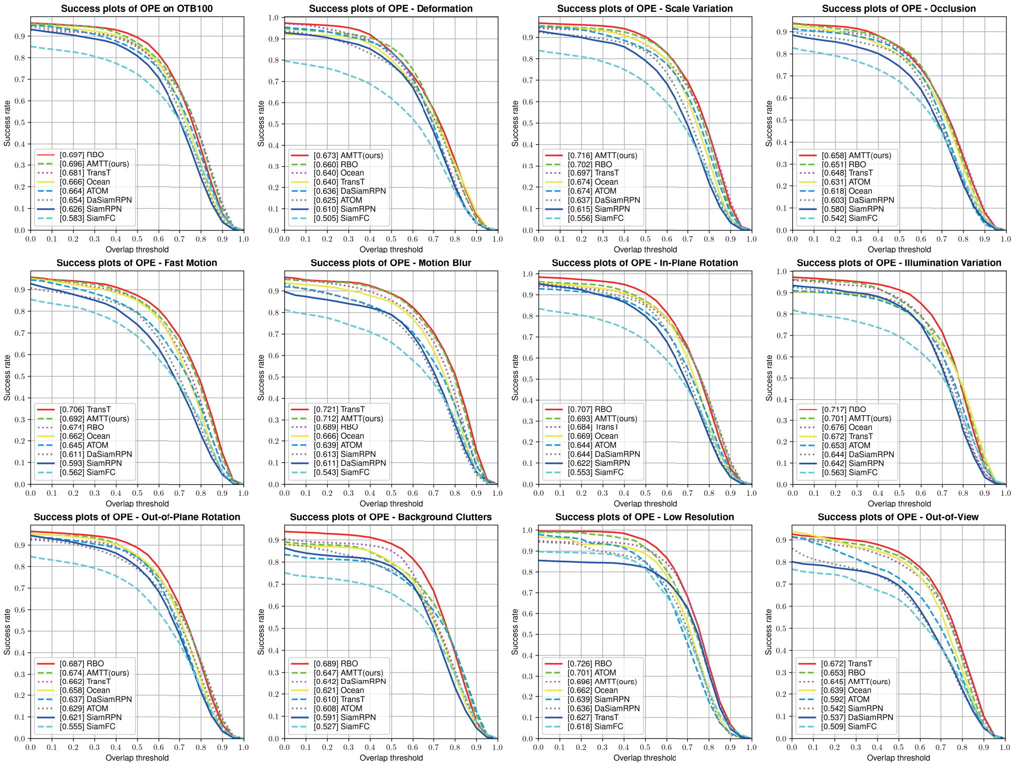

Figure 10. The figure shows the success rate of our method and other methods under multiple challenges on the OTB100 dataset.

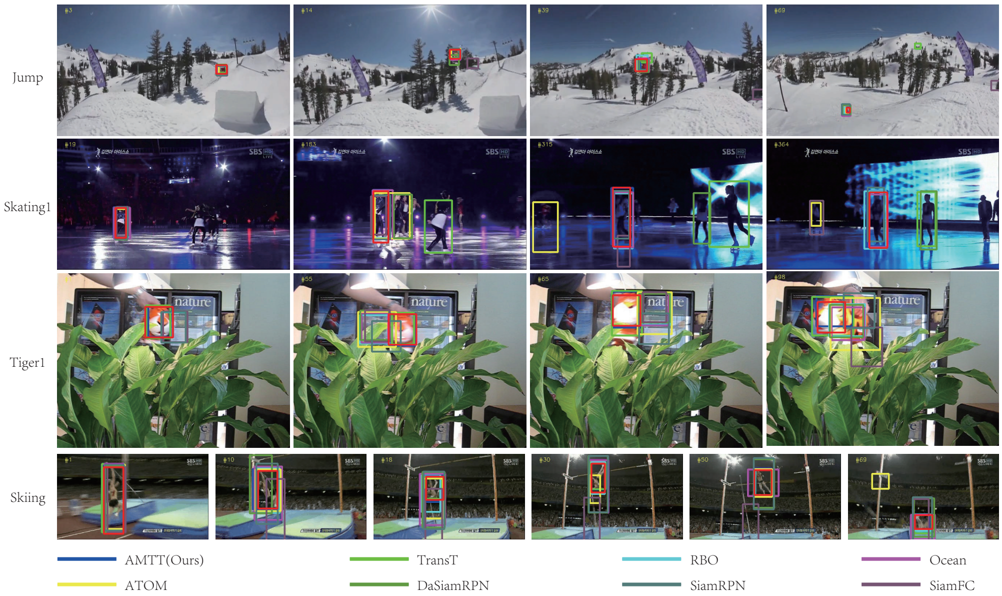

Figure 11. The figure shows some of the results of AMTT with other methods on the OTB100 dataset for three video sequences of Jump, Skating1, Tiger1 and Skiing. It is best viewed zoomed in, where the red box indicates the real box.

GOT-10k: GOT-10k is a training set containing 10,000 video sequences; 180 sequences are used as tests, and the training and test dataset categories do not overlap with each other. We validate our tracking methods by strictly adhering to the test protocol and submitting the test results to the official website for validation [14]. As shown in Table 3, we compare the average overlap (AO) and success rate ($SR_{0.5}, SR_{0.75}$). Our method demonstrates significant improvements over both classical methods and the advanced methods developed in recent years. There is a 4.8% improvement over GdaTFT [34] on AO and a 1.7% on $SR_{0.5}$. Compared with CIA [35], which also uses multilevel features, we lead by 0.9% on AO, indicating that our proposed algorithm can better combine multi-scale features and the tracking algorithm performs better.

Table 3. The table shows the results on the GOT-10k test set. The best two results are shown in red and blue fonts.

	ATOM [33]	**AutoMatch [36]**	**SiamGAT [3]**	**TrDiMP [37]**	**RBO [31]**	**UTT [38]**	**CIA [35]**	**GadTFT [34]**	**AMTT Ours**
AO (%)	55.6	65.2	62.7	67.1	64.4	67.2	67.9	65.0	69.8
$SR_{0.5}$ (%)	63.4	76.6	74.3	77.7	76.7	76.3	79.0	77.8	79.5
$SR_{0.75}$ (%)	40.2	54.3	48.8	58.3	50.9	60.5	60.3	53.7	63.4

TrackingNet: TrackingNet is a comprehensive object tracking dataset featuring a large training set and a test set that includes 511 video sequences with a variety of object classes [15]. Similar to GOT-10k, TrackingNet uses an online official test method. Our method also achieves commendable results when compared to recent methods. Table 4 presents the results on the TrackingNet test set, highlighting the effectiveness of our approach.

Table 4. The table shows the results on the TrackingNet test set. The best two results are shown in red and blue fonts.

	ATOM [33]	**SiamRPN++ [11]**	**AutoMatch [36]**	**SiamRCR [39]**	**TransT [10]**	**UTT [38]**	**CIA [35]**	**GadTFT [34]**	**AMTT Ours**
Prec. (%)	64.8	69.4	72.6	71.6	80.3	77.0	75.1	75.4	77.4
N.Prec. (%)	77.1	80.0	-	81.8	86.7	-	84.5	-	84.8
Success (%)	70.3	73.3	76.0	76.4	81.4	79.7	79.2	77.8	80.0

UAV123: UAV123 is a benchmark consisting of videos captured by UAVs which includes 123 sequences with an average of 915 frames each, containing 12 challenges of Aspect Ratio Change, Background Clutter, Camera Motion, Fast Motion, Full Occlusion, Illumination Variation, Low Resolution, Out-of-View, Partial Occlusion, Similar Object, Scale Variation and Viewpoint Change [17]. Table 5 shows the results of our algorithm in terms of accuracy and success rate in comparison with classical algorithms and advanced algorithms in recent years, while in Figure 12 the results are shown more clearly by means of curves. From the results, it can be seen that our algorithm achieves the best performance, and on the success rate graph, we improve by 1.7% compared to the second-ranked TrDiMP [37] and by 9.5% compared to HiFT [26], which also uses multi-scale features. This demonstrates that our proposed multi-scale feature fusion network effectively leverages multi-scale information, resulting in superior tracking performance. Additionally, Figure 13 compares our method with the top five state-of-the-art methods across various challenges, revealing that our approach consistently achieves the best results in nearly every scenario. Meanwhile, we visualize the partial tracking of the top five methods in three video sequences on the UAV123 dataset in Figure 14, and even if the object to be tracked is small, our tracker can accurately track the target.

Table 5. The table shows the results on the UAV123 dataset. The best two results are shown in red and blue fonts.

	Success (%)	**Precision (%)**
AMTT (Ours)	69.2	85.9
TrDiMP [37]	67.5	85.6
TransT [10]	67.0	85.2
SiamAttn [40]	65.9	83.9
STMTrack [41]	65.7	83.2
SiamPW-RBO [31]	65.6	82.6
AutoMatch [36]	65.4	81.3
Ocean [32]	63.1	81.1
SiamBAN [12]	61.2	78.0
HiFT [26]	59.7	77.1

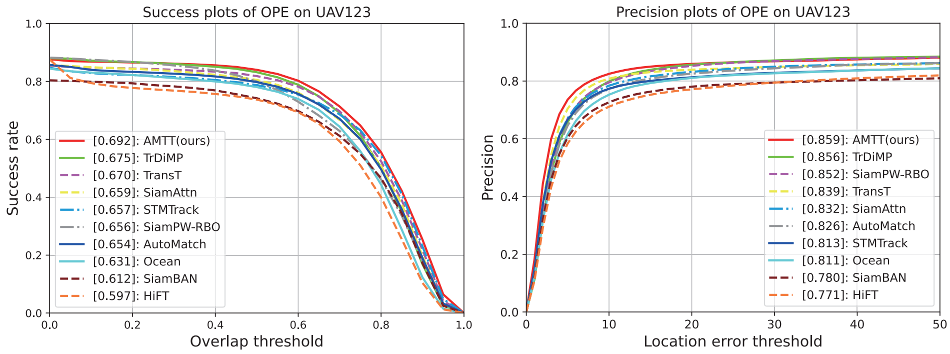

Figure 12. Results of our proposed AMTT with state-of-the-art methods on the UAV123 dataset in terms of precision and success rate.

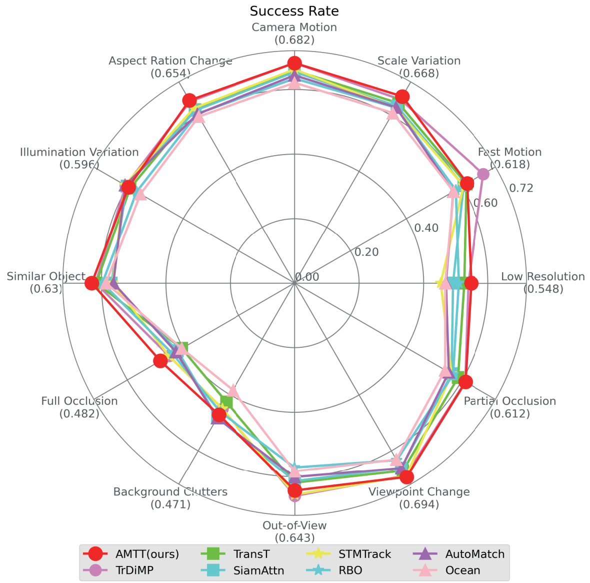

Figure 13. The results of the top eight success rate across challenges on the UAV123 dataset.

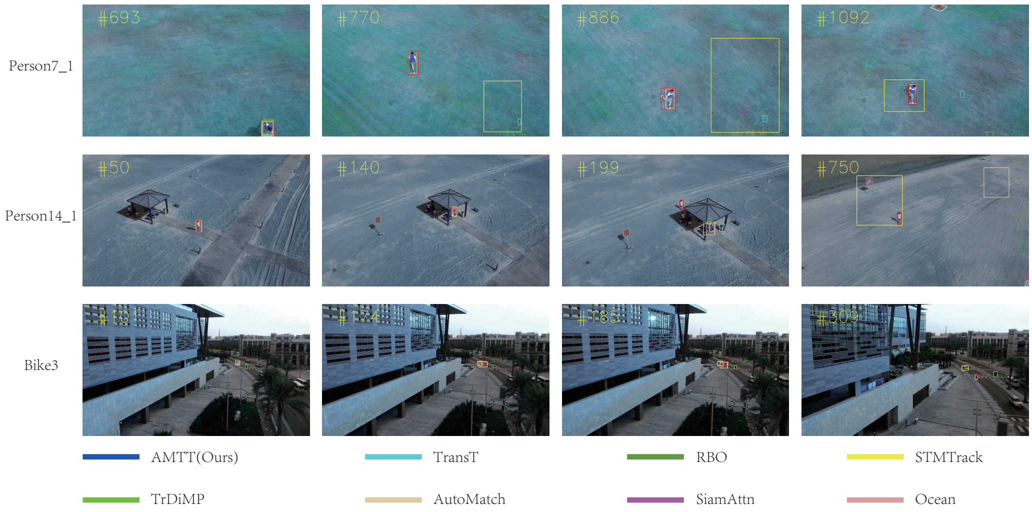

Figure 14. The figure shows some of the results of AMTT with other methods on the UAV123 dataset for three video sequences of Person7_1, Person14_1 and Bike3. It is best viewed zoomed in, where the red box indicates the real box.

4.3. Ablation Experiment

In this section, we perform ablation experiments on the proposed modules to check the performance of each module and validate it on the UAV123 dataset. First, we introduce the meaning of each abbreviation in Table 6. I denotes a multi-scale encoder using ResNet50 as a feature extraction network, an ordinary multicapitate attention composition and a single-scale decoder using multicapitate attention composition. II, III, IV, V are the codes for ablation experiments performed on I, respectively. MFE denotes the use of a multi-scale feature encoder of our design, FFB denotes the Feature Focusing Block and AD denotes the Anchor based decoder.

As can be seen from the first two rows of data in Table 6, I uses the original multi-head attention module as a multi-scale encoder, which has a lower performance of the tracker. II is the change of the encoder to the multi-scale feature encoder of our design; the accuracy of the tracker increases by 14.4% points. It shows that our designed encoder can better fuse multi-scale information. The traditional multihead attention module fuses all the multi-scale information, and the large amount of background information increases the computation of the module and pollutes the fused image features. In contrast, our designed decoder only takes the k most relevant locations in each layer for feature encoding, which is reflected in the feature map as the local features are continuously enhanced, and the background and interference information at the edges do not interact with the tracking object which is continuously suppressed.

Table 6. The table shows the results of ablation experiments on the UAV123 dataset for each module of our design.

	MFE	FFB	AD	Success	Precision
I				50.2	70.3
II	✓			$65.7_{15.5\%\uparrow}$	$84.7_{14.4\%\uparrow}$
III	✓	✓		$67.1_{1.4\%\uparrow}$	$86.3_{1.6\%\uparrow}$
IV	✓		✓	$66.8_{1.1\%\uparrow}$	$85.7_{1.0\%\uparrow}$
V	✓	✓	✓	$69.2_{3.5\%\uparrow}$	$85.9_{1.2\%\uparrow}$

The performances of the FFB module and the AD module are verified separately based on the inclusion of our designed encoder. As can be seen from the data in the second and third rows of Table 6, the performance of the tracker improves slightly after the introduction of the FFB module, thanks to the fact that after encoding the features, if the low-level features are directly discarded and the high-level features are directly decoded, there is a problem of information loss. We add the FFB module which can further enhance the high-level features before decoding to obtain the effect of information aggregation. Meanwhile, on the basis of adding the MFE module, we introduce the AD module. The data in the second and fourth rows show that there is also a small increase in the performance of the tracker. According to the analysis, the decoder after introducing the anchor information is able to produce finer results by guiding the decoder toward the correct object position and scale information through the pre-set Anchor. The last row is our designed AMTT tracking method, which introduces both FFB and AD modules on the basis of adding the MFE module, and from the ablation experiment, we can obtain the best result of our designed tracker. The baseline and the score of our method in the final prediction are visualized in Figure 15 using a heat map. It can be seen that our method is able to achieve more accurate tracking when there are similar objects with different scales in contact, and AMTT is able to clearly separate the tracked objects.

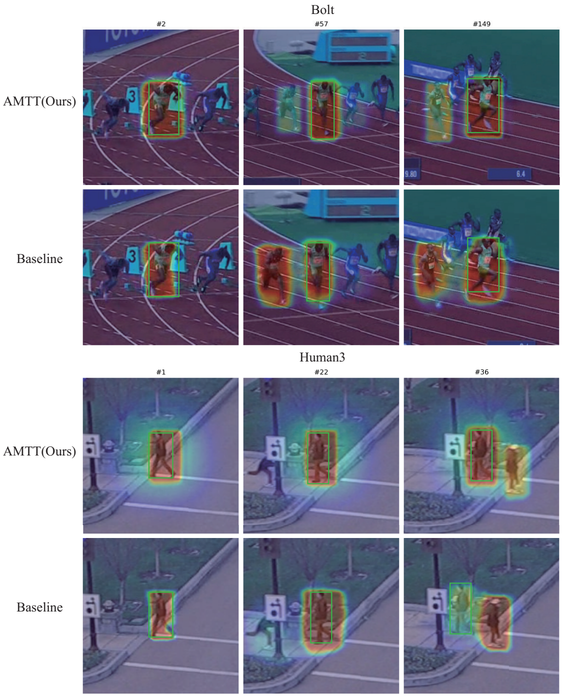

Figure 15. The figure shows the AMTT and baseline in the final prediction scores using a heat map. The green box represents the real object and the higher red color of the heat map represents the position where the network has a higher likelihood of predicting the object.

5. Conclusions

In this work, we propose an end-to-end anchor-based multi-scale transformer tracking method. Unlike existing feature fusion methods, we design a simple and comprehensive multi-scale feature fusion network. First, we design a deformable attention-based multi-scale feature encoder which suppresses the background information in the features through a self-enhancement module and reinforces the target information in the features through a cross-enhancement module, thus realizing efficient fusion of multi-scale features. Then, we propose the feature focusing block module to compress the number of encoded search features and encoded template features so as to reduce the decoding operations without loss of information. Finally, the focused features are decoded by an anchor-based decoder that utilizes feature anchor to guide the decoder to decode finer target locations. Our method is validated on the UAV123, OTB100, GOT10k and TrackingNet datasets. A success rate of 69.2% is achieved on the UAV123 dataset, demonstrating that our method can effectively fuse multi-scale features. Moreover, our method also achieves a leading position on several challenges of UAV123 and OTB100, such as scale transformations, fast moving object movement, and so on. Even if the object is deformed and its position changes too fast, our method can still track it well thanks to the proposed anchor-based decoder.

The proposed method can be applied to several practical application scenarios such as UAV tracking and video surveillance, and can better cope with the localization and tracking of targets at different scales. However, the method still has certain shortcomings. Compared with the ATOM method, the method does not introduce an update template to realize tracking, and it cannot avoid the situation of tracking failure due to the loss of target in long-time tracking. Therefore, in the future, an effective template updating method needs to be investigated and embedded into our method to realize target tracking adapted to different time lengths.

Author Contributions: Writing—original draft, software, Y.Z.; conceptualization and methodology, Y.Z. and Q.X.; visualization, Y.Z., Q.X. and N.L.; writing—review and editing, Y.Z., N.L. and H.D.; resources, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Our datasets are all from the public domain: TrackingNet [15], GOT-10k [14], LaSOT [29], COCO [30], OTB100 [16], UAV123 [17]. The original contributions presented in the study are included in the article, and further inquiries can be directed to the corresponding author.

Acknowledgments: We are grateful to the High-Performance Computing Center of Central South University for the assistance with the computations.

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

References

1. Jha, S.; Seo, C.; Yang, E.; Joshi, G.P. Real time object detection and trackingsystem for video surveillance system. *Multimed. Tools Appl.* **2021**, *80*, 3981–3996. [CrossRef]
2. Pereira, R.; Carvalho, G.; Garrote, L.; Nunes, U.J. Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Appl. Sci.* **2022**, *12*, 1319. [CrossRef]
3. Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; Shen, C. Graph attention tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9543–9552.
4. Voigtlaender, P.; Luiten, J.; Torr, P.H.S.; Leibe, B. Siam R-CNN: Visual Tracking by Re-Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6577–6587.
5. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9914, pp. 850–865.
6. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
7. Fan, H.; Ling, H. Siamese cascaded region proposal 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. 7952–7961.
8. 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.
9. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6268–6276.
10. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8122–8131.
11. 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4277–4286.
12. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13–19 June 2020.
13. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
14. 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]
15. 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.
16. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
17. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
19. Hao, S.; Gao, S.; Ma, X.; An, B.; He, T. Anchor-free infrared pedestrian detection based on cross-scale feature fusion and hierarchical attention mechanism. *Infrared Phys. Technol.* **2023**, *131*, 104660. [CrossRef]
20. Hao, S.; An, B.; Ma, X.; Sun, X.; He, T.; Sun, S. PKAMNet: A transmission line insulator parallel-gap fault detection network based on prior knowledge transfer and attention mechanism. *IEEE Trans. Power Deliv.* **2023**, *38*, 3387–3397. [CrossRef]

21. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
22. Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; Wang, J. Conditional detr for fast training convergence. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3651–3660.
23. 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]
24. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9909, pp. 472–488.
25. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.
26. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. Hift: Hierarchical feature transformer for aerial tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, New Orleans, LA, USA, 18–24 June 2021; pp. 15457–15466.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 658–666.
29. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. Lasot: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5374–5383.
30. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer International Publishing: Berlin/Heidelberg, Germany, 2014.
31. Tang, F.; Ling, Q. Ranking-based Siamese visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8741–8750.
32. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 771–787.
33. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4655–4664.
34. Liang, Y.; Li, Q.; Long, F. Global dilated attention and target focusing network for robust tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1549–1557.
35. Pi, Z.; Wan, W.; Sun, C.; Gao, C.; Sang, N.; Li, C. Hierarchical feature embedding for visual tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2022; pp. 428–445.
36. Zhang, Z.; Liu, Y.; Wang, X.; Li, B.; Hu, W. Learn to match: Automatic matching network design for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 13339–13348.
37. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1571–1580.
38. Ma, F.; Shou, M.Z.; Zhu, L.; Fan, H.; Xu, Y.; Yang, Y.; Yan, Z. Unified transformer tracker for object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8781–8790.
39. Peng, J.; Jiang, Z.; Gu, Y.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Lin, W. Siamrcr: Reciprocal classification and regression for visual object tracking. *arXiv* **2021**, arXiv:2105.11237.
40. 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, 14–19 June 2020; pp. 6728–6737.
41. Fu, Z.; Liu, Q.; Fu, Z.; Wang, Y. Stmtrack: Template-free visual tracking with space-time memory networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13774–13783.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.