drones

MDPI

Article

Learning to Propose and Refine for Accurate and Robust Tracking via an Alignment Convolution

Zhiyi Mo [1,2] **and Zhi Li** [1,*]

[1] The Guangxi Key Laboratory of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China; zhiyim@gxuwz.edu.cn

[2] Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou 543002, China

[*] Correspondence: zhili@gxnu.edu.cn

Abstract: Precise and robust feature extraction plays a key role in high-performance tracking to analyse the videos from drones, surveillance and automatic driving, etc. However, most existing Siamese network-based trackers mainly focus on constructing complicated network models and refinement strategies, while using comparatively simple and heuristic conventional or deformable convolutions to extract features from the sampling positions that may be far away from a target region. Consequently, the coarsely extracted features may introduce background noise and degrade the tracking performance. To address this issue, we present a propose-and-refine tracker (PRTracker) that combines anchor-free style proposals at the coarse level, and alignment convolution-driven refinement at the fine level. Specifically, at the coarse level, we design an anchor-free model to effectively generate proposals that provide more reliable interested regions for further verifying. At the fine level, an alignment convolution-based refinement strategy is adopted to improve the convolutional sampling positions of the proposals, thus making the classification and regression of them more accurate. Through using alignment convolution, the convolution sampling positions of the proposals can be efficiently and effectively re-localized, thus improving the accuracy of the extracted features. Finally, a simple yet robust target mask is designed to make full use of the initial state of a target to further improve the tracking performance. The proposed PRTracker achieves a competitive performance against six tracking benchmarks (i.e., UAV123, VOT2018, VOT2019, OTB100, NfS and LaSOT) at 75 FPS.

Keywords: accurate tracking; robust tracking; anchor-free; alignment convolution; propose-and-refine

check for updates

Citation: Mo , Z.; Li, Z. Learning to Propose and Refine for Accurate and Robust Tracking via an Alignment Convolution. *Drones* **2023**, *7*, 343. https://doi.org/10.3390/drones7060343

Academic Editor: Anastasios Dimou

Received: 24 April 2023
Revised: 16 May 2023
Accepted: 19 May 2023
Published: 25 May 2023

1. Introduction

Given the initial state of a target object, visual tracking aims to estimate the state (usually represented by a bounding box) of the target in each frame of a video sequence. Accurate and robust tracking is required by various practical applications such as intelligent drones for urban monitoring [1], computer interactions [2], automatic driving [3] and video surveillance [4]. Recently, although numerous top-performing trackers [5–14] have been proposed, it is still quite difficult to achieve accurate and robust tracking in dynamically complicated scenes containing similar objects, non-rigid object deformation, background clutters and fast motion. As shown in Figure 1 and Table 1, through empirical analysis, we find that the accuracy of features extracted in each tracker is greatly affected by convolutional sampling locations in different convolution operations, e.g., conventional [15], deformable [16], and alignment convolutions. The conventional convolution utilizes regular sampling grids to extract the features. When the features are used in a tracker [5], it cannot handle large-scale and deformation changes. As the deformable convolution adds 2D learnable offsets to the regular grid sampling locations, it is robust to geometric transformations. However, when it is used in a tracker [17], it still introduces interference noises due to the lack of supervision information. Consequently, one critical obstacle is how to extract precise and robust features to accurately describe the target while discriminating it from the backgrounds. To address this issue, most existing methods focus on the two aspects of building an accurate and robust tracker, i.e., feature learning and coarse-to-fine refinement.

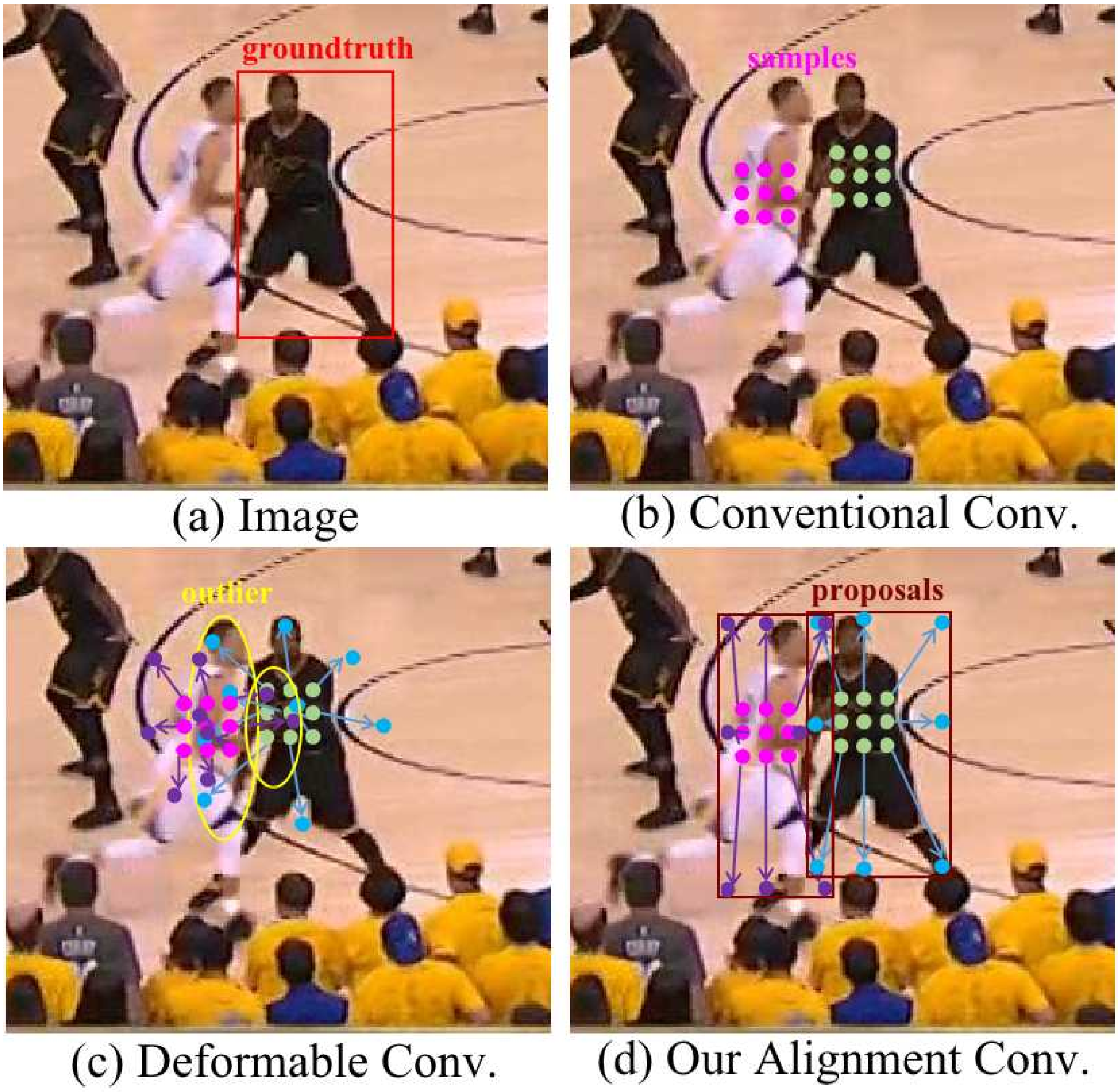


Figure 1. Three typical convolutional methods used to extract features in each tracker: (**a**) An image; (**b**) using a conventional convolution to extract features. The extracted features are inaccurate to deal with deformation changes due to regular sampling grids. (**c**) Using a deformable convolution to extract features. This uses learnable offsets to obtain sampling points; however, this may brings outliers instead. (**d**) Using the proposed alignment convolution with the supervision of proposals to extract features. The convolutional sampling points can be effectively re-localized to make the classification and regression of proposals more accurate and robust. See Section 4 for empirical results of the three strategies.

Feature learning: to construct a robust feature representation, classical Siamese network-based trackers [5,18,19] use large-scale offline training samples to train a two-flow deep Siamese network. However, they merely rely on a conventional convolution to extract features and a heuristic multi-scale searching mechanism to estimate the location and size of a target, respectively. These basic yet key operations may introduce background noise. As a result, they suffer from inaccurate feature extraction, and thus cannot accurately predict the target deformations. To improve the accuracy, recent trackers explore anchor-based and -free solutions from different aspects. The most representative anchor-based trackers are SiamRPN [20] and its follow-up work (e.g., DaSiamRPN [21] and SiamRPN++ [9]). SiamRPN introduces a regional proposal network (RPN) [22] for similarity matching and bounding box regression. DaSiamRPN [21] and SiamRPN++ [9] improve the robustness of SiamRPN through distractor-aware training and a deeper network, respectively. Although these anchor-based trackers can handle changes in the scale and aspect ratio of the targets, they are sensitive to the numbers, sizes and aspect ratios of anchor boxes. In addition, because the scale and aspect ratio of the anchor boxes are fixed, even if heuristics are used to adjust the parameters, they are still difficult to handle objects with large shape and pose variations. To address this issue, the anchor-free methodology [6] is introduced for visual tracking. The simple yet effective anchor-free trackers [6,23,24] directly classify objects and regress their bounding boxes from pre-defined positions, thus avoiding the design of the anchor boxes. They use the classification information from the pre-defined locations to choose the optimal bounding box. However, most of the pre-defined positions may be far away from the target areas. Consequently, their tracking accuracy is still degraded due to inaccurate feature extraction from coarse regression boxes and conventional convolutions.

Coarse-to-fine refinement: some Siamese-based trackers rely on a coarse-to-fine paradigm to balance accuracy and robustness. Representative trackers include Siam R-CNN [7] and SPM [10] that are composed of a coarse matching stage and a refinement stage. In the refinement stage, they use the RoI operators to extract the features of the interested regions for classification and regression, such as RoIAlign [25] and PrPool [26]. Compared with the above trackers (e.g, SiamRPN [20], SiamRPN++ [9], SiamBAN [6]) directly predicting a regression box, Siam R-CNN [7] and SPM [10] have higher accuracy due to the refinement strategies. However, one limitation of Siam R-CNN [7] is the complex model that needs a cascaded scheme to ensure detection accuracy. As a result, it runs very slowly, i.e., less than 5 FPS. SPM [10] depends on the candidate results from the coarse results to trigger its refining process. However, effectively generating the better candidate results that facilitate the refining process is crucial. Moreover, despite their success, Siam R-CNN [7] and SPM [10] are limited by the fact that they cannot extract accurate features due to using conventional convolutions with fixed sampling positions.

In summary, for feature learning-based trackers [6,9,20], such approaches either rely on anchor-based mechanism or utilize anchor-free mechanism to achieve tracking target. However, they are sensitive to the numbers, sizes and aspect ratios of anchor boxes, or the pre-defined positions may be far from the target area. Therefore, their tracking accuracy is still degraded due to inaccurate feature extraction from coarse regression boxes and conventional convolutions. For coarse-to-fine refinement-based trackers [7,10], although such methods obtain good performance, they are limited by the complexity of the model and they are not efficient in generating proposals.

In this paper, inspired by the observations that feature extraction is crucial and the coarse-to-fine mechanism is a powerful tool, we propose a novel propose-and-refine tracker (PRTracker) for accurate and robust tracking. The proposed PRTracker effectively fuses anchor-free style proposals at the coarse level, and alignment convolution-driven refinement at the fine level. At the coarse level, an anchor-free model is utilized to effectively generate proposals that provide more reliable interested regions, while discarding the majority of backgrounds. At the fine level, an alignment convolution-based refinement strategy is adopted to improve the convolution sampling positions of the proposals, thus making the classification and regression of the proposals more accurate. The initial proposals generated at the coarse level are used as supervised information to make the sampling grids more stable. The benefit from alignment convolution is that the convolution sampling positions of the proposals can be efficiently and effectively re-localized, thus improving the accuracy of extracted features. Finally, we design a simple yet robust target mask to make full use of the initial state of the targets to further improve the tracking performance. To sum up, the main contributions of this work are three-fold:

- The paper explores a major challenge that leads to inaccurate target localization, while often not discussed in the tracking literature. Based on careful investigation, this paper discovers the inaccurate convolution sampling points likely to lead to incorrect feature extraction, which degrades a tracker.
- The paper designs a simple yet efficient propose-and-refine mechanism that is driven by an alignment convolution to classify and refine the proposals. By naturally accentuating the advantages of each component, the proposed PRTracker can not only effectively obtain reliable proposals, but also provide more accurate and robust features for further classification and regression.
- The paper extensively validates the proposed PRTracker against six benchmarks including LaSOT [27], VOT2018 [28], VOT2019 [29], NfS [30], OTB100 [31] and UAV123 [32]. The results show the accuracy and robustness of the proposed PRTracker.

The rest of this paper is organized as follows: we briefly present an overview of the related work in Section 2. The framework of the proposed PRTracker is described in detail in Section 3. The experimental results and corresponding analyses are systematically shown in Section 4. Finally, we conclude the paper in Section 5.

Table 1. Illustrations of the different properties of the existing convolution-based trackers.

Types	Sample Mode	Coarse-to-Fine Refinement
Conventional Convolution-Based Trackers [6,9,20]	Regular Grid Sampling	No
Deformable Convolution-Based Trackers [17]	Learnable Offset Sampling	No
Alignment Convolution-Based Trackers (Proposed PRTracker)	Learnable Offset Sampling with the Proposal Supervision Signal	Yes

2. Related Work

With the rapid development of deep learning, many excellent works [5,9,10,20,21,25,33–37] have emerged in the research community of objecting tracking. This section provides a brief overview of the related methods, i.e., coarse target localization and coarse-to-fine localization in object tracking.

2.1. Coarse Target Localization in Object Tracking

Multi-scale searching mechanism-based trackers: the trackers based on multi-scale searching mechanisms typically rely on a simple and heuristic-scale pyramid to estimate the location and size of a target. Representative approaches include discriminative correlation filter (DCF)-based trackers [33,38] and Siamese network-based trackers [5]. By revisiting the core DCF formulation, ECO [33] introduces a factorized convolution operator, a compact generative model, and a conservative model update strategy into DCF, respectively. Consequently, it improves the spatial-temporal efficiency based on a C-COT model [38]. SiamFC [5] uses a new fully convolutional Siamese network as a basic tracking model. It exceeds the real-time requirement. However, the scale pyramid-based test makes these trackers inflexible to accurately estimate the scale and aspect ratio of a target.

Anchor-based trackers: anchor-based trackers treat a tracking task as a classification-regression problem, in which the bounding box coordinates and target–background probabilities are directly predicted. One popular anchor-based tracker is SiamRPN [20] which introduces a region proposal network (RPN) into a Siamese network-based tracker. Through directing classification and regression anchors, SiamRPN avoids the multi-scale search and achieves accurate results. Inspired by the success of SiamRPN, numerous anchor-based trackers have been proposed. DaSiamRPN [21] adopts a distractor-aware feature-learning scheme to promote the discriminative power of its network. SiamRPN++ [9], SiamMask [35] and SiamDW [34] introduce modern deep neural networks into anchor-based trackers to improve their tracking performance, such as ResNet [39], ResNeXt [40], and MobileNet [41]. Although anchor-based trackers [9,10,20] can handle changes in scale and aspect ratio, it is necessary to carefully design anchor boxes based on heuristic knowledge, which introduces many hyperparameters and computational complexity.

Anchor-free trackers: recently, a few of anchor-free trackers [6,8,23] have been proposed. Different from anchor-based trackers, the anchor-free trackers treat each pixel as an anchor point, and thus can directly predict the position of a target. In contrast to anchor-based trackers, anchor-free trackers avoid hyperparameters associated with the anchor boxes and is more flexible and general. However, the above anchor-free trackers have the problem of misalignment between the predicted bounding boxes and the sampling features. Most of the pre-defined positions may be far away from the target areas. Consequently, their tracking accuracy is still degraded due to inaccurate feature extraction from coarse regression boxes and conventional convolutions.

2.2. Coarse-to-Fine Localization in Object Tracking

Recently, numerous top-performing trackers [7,10,42,43] rely on a coarse-to-fine strategy consisting of a coarse and refinement stage to gradually obtain accurate and robust tracking results. Generally speaking, they firstly predict the coarse target states at a coarse stage. Then, the coarse target states are adjusted at a refinement stage. In ATOM [43], an online classification module is firstly utilized to estimate the coarse target positions. Meanwhile, some samples around the coarse target positions are randomly generated. At the refinement stage, more precise bounding boxes are obtained by maximizing the overlap between ground truth and these samples. In SPM [10], candidate results generated by a coarse stage are utilized to crop the ROI regions for further refining. However, the ROI regions may be inaccurate if the coarse results are unreliable. In contrast, we employ coarse results as the initial results. Then, we further adjust the convolution sampling points to obtain accurate features before classification and regression. Siam R-CNN [7] improves its accuracy by maintaining a tracklet and matching the similarity between coarse results and a template. The refinement strategy of Siam R-CNN is similar to post-processing, which is complex and computationally intensive. Meanwhile, in other fields, some new strategies have been proposed in some of the work to improve the video understanding capabilities. For example, Gomaa et al. [44] used top- and bottom-hat transformations aided by the morphological operation to capture the target object in the detection phase, and then performed motion feature point analysis using a combined technique between KLT tracker and K-means clustering to achieve a better decision result. Based on a YOLOv2 algorithm, Gomaa et al. [45] used a two-stage strategy of detection and tracking to implement the feature point motion analysis and vehicle detection and counting method. Chang et al. [46] explored a new convolution autoencoder architecture that can dissociate the spatio-temporal representation to separately capture spatial and temporal information, thereby improving the detection performance of fast moving outliers. In contrast, we found that the inaccurate convolution sampling points in the target regions are the main challenge that greatly degrades the classification and regression results. Therefore, this paper introduces an alignment convolution to adaptively re-localize the convolution sampling points, thus refining the classification and regression results. Based on a plug-and-play style refinement module, Alpha-Refine [42] is able to efficiently refine a base tracker's outputs. However, its tracking performance may be limited by the base tracker. By end-to-end training, the proposed PRTracker can effectively optimize the coarse and refinement stages together.

2.3. Feature Alignment in Object Detection

Feature alignment usually refers to the alignment between convolutional features and proposals/RoIs, which is very important for object detection and localization. In two-stage object detection, the RoI operators (e.g., RoIPooling [47], RoIAlign [25], PrPool [26] and Deformable RoIPooling [16]) are usually used to extract fixed-length features inside the RoIs. The fixed-length features can effectively represent the object appearances. In one-stage detectors [48,49], an important requirement is to maintain a full convolutional structure, improving the processing speed. However, the used RoI operators cannot meet this requirement due to the introduction of fully connected layers. Recently, some detectors [50,51] used a deformable convolution to achieve feature alignment. Their offset values are usually obtained by calculating the offsets between a pre-defined anchor box and a refined anchor box. Although the deformable convolution can change the convolution sampling points, it lacks explicit supervision information for effectively training. Thus, the obtained convolution sampling points may be unstable. Ocean [8] adopts a feature alignment schema that is independent of the classification results for tracking. It directly extracts object-aware features from the estimated bounding boxes, without considering the classification scores. In this work, we designed an alignment convolution to re-localize the convolution sampling points, thus capturing more accurate features. To effectively adjust the convolution sampling points, we take the coarse results generated at the coarse stage as the supervision information during training. At the refinement stage, the convolution sampling points can be effectively refined based on the coarse results.

3. The Proposed Tracker

In this section, we develop a novel tracking method, PRTracker, that learns to propose and refine for accurate and robust tracking via an alignment convolution. As shown in Figure 2, the PRTracker consists of a Siamese network backbone, a propose-and-refine module and a target mask. The shared parameters of the Siamese backbone network are responsible for extracting features from both a template and search patch. Meanwhile, we provide the initial state of a target to the template features through the target mask. After this, the template and search features are introduced into the DW-Corr module for learning the target representation. Finally, we use the learned features as inputs to the propose-and-refine module for target localization. Specifically, the propose-and-refine module first generates proposals at the coarse level. Then, it classifies and refines them through an alignment convolution at the fine level. Below we give a detailed description of each component in the framework, and then summarize the proposed PRTracker.

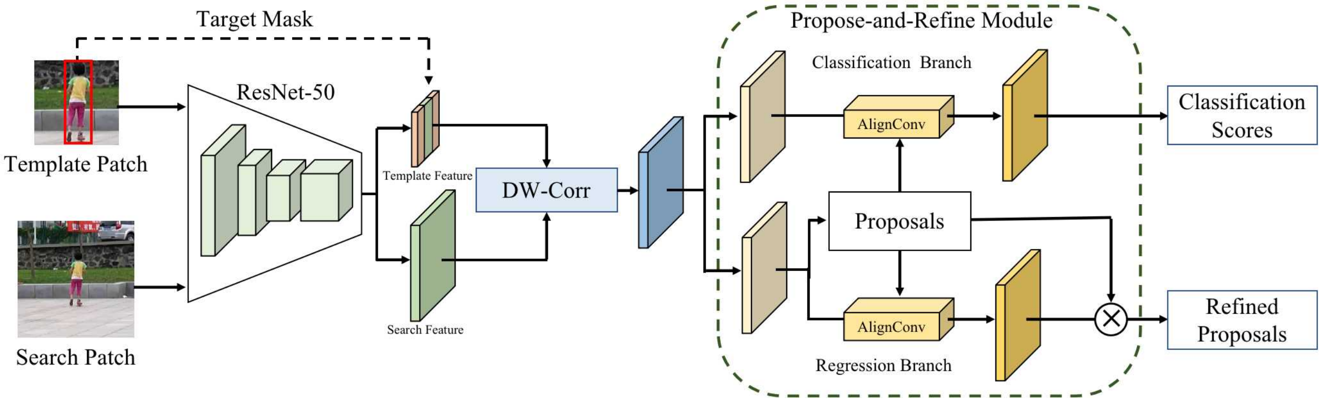


Figure 2. The framework of the proposed PRTracker. First, we extract template and search features using a ResNet-50 convolutional neural network. Next, we feed the template and search features into the DW-Corr module for learning the target representation. Finally, we use the learned features as inputs to the propose-and-refine module for target localization. The DW-Corr represents the depth-wise cross-correlation operation and the AlignConv represents the alignment convolution. The propose-and-refine module contains two sub-networks, one to generate and refine proposals, and the other to classify proposals.

3.1. The Siamese Network Backbone

The goal of the Siamese network backbone is to extract features from both a template and search patch. As is well known, modern deep neural networks [39–41] have been proven to be effective in Siamese network-based trackers [9,34,35]. To clearly show the effectiveness of the propose-and-refine module, we use a modified ResNet-50 [39] as the backbone network and extract features from the *conv*4 block. To increase the resolution of the feature maps, we remove the convolution stride of the *conv*4 block. At the same time, to maintain the receptive fields, the atrous rate of all 3 × 3 convolutions of the *conv*4 block is set to 2. The Siamese network backbone takes a pair of pictures as the input, i.e., a template patch (denoted as z), and a search patch (denoted as x). Table 2 shows the details of the Siamese network backbone.

Table 2. The details of the Siamese network backbone in the proposed framework. Details of each building block, template, and search features are shown in square brackets.

Block	Backbone	Search Branch Output Size	Template Branch Output Size
conv1	$7 \times 7, 64$, stride 2	125×125	61×61
conv2_x	3×3 max pool, stride 2 $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	63×63	31×31
conv3_x	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	31×31	15×15
conv4_x	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	31×31	15×15
adjust	$1 \times 1, 256$	31×31	7×7
xcorr	depth-wise	25×25	

3.2. Alignment Convolution

Extracting accurate and robust features is a challenging and important problem in visual tracking. As shown in Figure 3, the conventional convolution samples the features at contiguous locations. If a target undergoes complicated appearance variations, the conventional convolution may introduce a lot of background interference information due to only extracting features with fixed sampling positions from a rectangular area. To focus on the ROI of a rectangular area, the deformable convolution changes the spatial sampling locations by learning the offsets. However, the convolution sampling positions of the deformable convolution are learnt without explicitly supervised information. To effectively re-localize the convolution sampling points, thus capturing more accurate features, we design an alignment convolution trained with explicitly supervised information for the propose-and-refine module. Specifically, given a bounding box and a position within the bounding box, the alignment convolution will sample from nine positions, i.e., one given position, four corner positions of the bounding box, and four vertical mapping positions of the position on the sides of the bounding box. For example, the given position is (x, y), and the distances from the position to the left, top, right, and bottom of the bounding box are l', t', r' and b', respectively. The nine sampling positions of the alignment convolution will be $(x-l', y-t')$, $(x, y-t')$, $(x+r', y-t')$, $(x-l', y)$, (x, y), $(x+r', y)$, $(x-l', y+b')$, $(x, y+b')$ and $(x+r', y+b')$, respectively. Therefore, given the offsets l', t', r' and b', the alignment convolution can be easily implemented with a deformable convolution with a given offset [16]. In comparison, the deformable convolution generates the learnable offsets through the convolution operation, making the sampling points random around the current location, and the alignment convolution is designed with the supervised information. Therefore, the alignment convolution can not only capture the geometric information, but also context information of the bounding box. It is worth noting that the computation of the offset values is not part of the convolutional computation, which comes from the first stage of the proposal generation process. Thus, the alignment convolution does not increase the computational burden compared with conventional convolution. Section 4.3 discussions more the effectiveness of the three different convolutions for the proposed tracker.

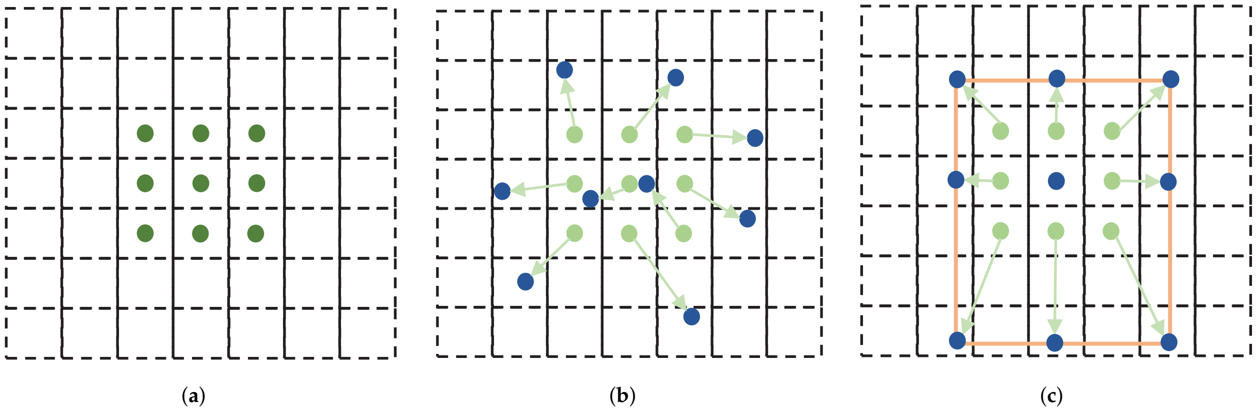


Figure 3. Illustration of the sampling locations in different convolutions with a 3 × 3 kernel. (**a**) The conventional convolution. (**b**) Deformable convolution [16]. (**c**) Proposed alignment convolution, which can effectively refine the convolution sampling positions and improve the accuracy of the extracted features.

3.3. The Propose-and-Refine Module

The propose-and-refine module proceeds by first obtaining a set of proposal solutions via an anchor-free model at the coarse level, and then optimizing them via an alignment convolution at the fine level. Specifically, as shown in Figure 2, the input of our propose-and-refine module is a combined feature map P that is generated by combining the template and search patch features through a depth-wise cross-correlation layer [9]:

$$P = \varphi(x) \star \varphi(z), \tag{1}$$

where $\varphi(x)$ and $\varphi(z)$ represent the template and search patch features, respectively. $\star$ denotes a convolution operation with $\varphi(z)$ as the convolution kernel. Then, we apply two non-shared sub-networks with three 3 × 3 convolution layers on the combined feature map P to obtain a classification and regression feature map. Furthermore, the regression sub-network generates proposals (l', t', r' and b') on the regression feature map through a 3 × 3 convolution layer and a 1 × 1 convolution layer. After obtaining the proposals, we use a 3 × 3 alignment convolution layer and a 1 × 1 convolution layer to classify and refine them on the classification and regression feature map, respectively. Given the proposals (l', t', r' and b'), an alignment convolution is used to sample on the classification feature map to predict the target–background scores. Meanwhile, another alignment convolution is used to sample the regression feature map to predict the four scale factors (dl, dt, dr and db). Finally, we can obtain the refined proposals, i.e., (l, t, r and b) = ($dl * l', dt * t', dr * r'$ and $db * b'$). Please note that we apply $exp(x)$ to map the regression output values to $(0, +\infty)$

since they are positive real numbers. Moreover, for each location on the classification or regression feature map, we can easily map it to the search patch. For example, the location (i, j) corresponds to $(p_i, p_j) = [\lfloor \frac{w_{im}}{2} \rfloor + (i - \lfloor \frac{w}{2} \rfloor) \times s, \lfloor \frac{h_{im}}{2} \rfloor + (j - \lfloor \frac{h}{2} \rfloor) \times s]$. w_{im} and h_{im} represent the width and height of the search patch, respectively. s represents the total stride of the network.

Based on our propose-and-refine module with an alignment convolution, the convolution sampling positions of the proposals can be efficiently and effectively re-localized, thus improving the accuracy of extracted features. Therefore, the proposed PRTracker can make the classification and regression of the proposals more accurate. Note that although the proposed PRTracker uses two regressions, only one depth-wise cross-correlation layer is required. In contrast, both SiamRPN++ [9] and SiamBAN [6] need six depth-wise cross-correlation operations. As a result, the proposed PRTracker is more efficient than previous anchor-based and -free trackers [6,8,9,23].

3.4. The Target Mask

To make full use of a ground truth bounding box from the initial frame, we design a target mask. As shown in Figure 2, the bounding box in the template patch is given to the template features through the target mask. Specifically, the target mask is generated according to the bounding box of a target. The values of the target mask are set to 1 if the corresponding positions are in the ground truth bounding box. Otherwise, they are set as -1. The basic idea of our target mask is to effectively use the background information. Thus, we set the values of the target mask to -1 instead of 0. Due to the ReLU activate function, there are no negative values in the template feature map before fusing the target template. Therefore, the template feature map is multiplied by the target mask to clearly distinguish the target and background features. Please note that this simple yet effective operation hardly adds computational overhead to the proposed PRTracker.

3.5. Ground Truth and Loss

Classification labels and regression targets. Inspired by SiamBAN [6], we utilize an ellipse figure region to design the labels. It is used for effective division of positive or negative samples. In detail, we first set an ellipse E_1 and E_2 according to a ground truth bounding box. The width, height, top-left corner, centre point and bottom-right corner of the ground truth bounding box are represented by g_w, g_h, (g_{x_1}, g_{y_1}), (g_{x_c}, g_{y_c}) and (g_{x_2}, g_{y_2}), respectively. The centre and axes length of ellipse E_1 are denoted by (g_{x_c}, g_{y_c}) and $\frac{g_w}{2}$, $\frac{g_h}{2}$, respectively. Ellipse E_1 is formulated by:

$$\frac{(p_i - g_{x_c})^2}{(\frac{g_w}{2})^2} + \frac{(p_j - g_{y_c})^2}{(\frac{g_h}{2})^2} = 1, \tag{2}$$

The centre and axes length of ellipse E_2 are denoted by (g_{x_c}, g_{y_c}) and $\frac{g_w}{4}$, $\frac{g_h}{4}$, respectively. Ellipse E_2 is formulated by:

$$\frac{(p_i - g_{x_c})^2}{(\frac{g_w}{4})^2} + \frac{(p_j - g_{y_c})^2}{(\frac{g_h}{4})^2} = 1, \tag{3}$$

During the proposal generation process, each position of the feature map inside the ground truth bounding box is utilized to train the regression network. After obtaining the proposals, we calculate their intersection over union (*IoU*) with the ground truth bounding box.

For classification labels, if the location (p_i, p_j) falls within ellipse E_2 and its *IoU* more than 0.6, it is assigned with a positive label. If it falls outside ellipse E_1 and its *IoU* less than 0.3, it is assigned with a negative label. Others are ignored.

In the proposal refinement stage, if the location (p_i, p_j) falls within the ground truth and its *IoU* more than 0.6, it needs to refine the proposals.

The proposal generation and refinement have the same targets, expressed as:

$$\begin{aligned} d_l &= p_i - g_{x_1}, \\ d_t &= p_j - g_{y_1}, \\ d_r &= g_{x_2} - p_i, \\ d_b &= g_{y_2} - p_j, \end{aligned} \tag{4}$$

where d_l, d_t, d_r and d_b are the offsets from the location to the four sides of a proposal.

Classification loss and regression loss. The multi-task loss function is formulated as follows:

$$L = \lambda_1 L_{cls} + \lambda_2 L_{gen} + \lambda_3 L_{ref}, \tag{5}$$

where L_{cls} is a cross-entropy loss used for proposals classification, and L_{gen} and L_{ref} are *IoU* losses used for proposal generation and refinement, respectively. We simply set $\lambda_1 = \lambda_2 = \lambda_3 = 1$. The *IoU* loss is defined as:

$$L_{IoU} = 1 - IoU, \tag{6}$$

where IoU represents the area ratio of *IoU* of the proposals and ground truth bounding box.

3.6. Training and Inference

Training. The training datasets include ImageNet VID [52], YouTube-BoundingBoxes [53], COCO [54], ImageNet DET [52], GOT10k [55] and LaSOT [27]. We extract image pairs from these videos or images, and crop the template and search patches from them. The sizes of the template and search patch are set as 127 × 127 and 255 × 255 pixels, respectively. For classification loss, we select 16 positive samples and 48 negative samples from each pair of images for training. For regression loss, the positive samples in each pair of images participate in training.

Inference. In the inference stage, we first crop a template patch from an initial frame. Then, we extract features and construct a target mask. For each subsequent frame, we crop a search patch based on the target position of the previous frame. Then, we extract the features of the search patch. Furthermore, we obtain the predicted values $(l, t, r$ and $b)$ through our propose-and-refine module. Therefore, the predicted bounding boxes can be obtained by the following equation:

$$\begin{aligned} p_{x_1} &= p_i - l, \\ p_{y_1} &= p_j - t, \\ p_{x_2} &= p_i + r, \\ p_{y_2} &= p_j + b, \end{aligned} \tag{7}$$

where $(p_{x_1},\ p_{y_1})$ and $(p_{x_2},\ p_{y_2})$ are the top-left corner and bottom-right corner of the prediction boxes, respectively.

After generating the predicted bounding boxes, we use a cosine window and scale change penalty schema to smooth the movements and scale changes of the target [20]. Finally, we select the predicted bounding box with the highest score, and update the bounding box in the previous frame through linear interpolation. Algorithm 1 summarizes the tracking process of the proposed PRTracker.

Algorithm 1: Accurate and robust tracking with PRTracker

```
Input: Frames $\{X_{t=1}^{T}\}$ and initial bounding box $b_1$ of $X_1$.
Output: Tracking results $\{b_{t=2}^{T}\}$.
Extract a target template $z$ in $X_1$ using $b_1$;
Extract features $\{\varphi(z)\}_{l=3}^{5}$ for $z$ from our model;
Obtain features $\left\{[\varphi^l(z)]_{cls}\right\}_{l=3}^{5}$ and $\left\{[\varphi^l(z)]_{reg}\right\}_{l=3}^{5}$ of $z$ from $\left\{\varphi^l(z)\right\}_{l=3}^{5}$ and add
  a target mask;
for $t = 2$ to $T$ do
    Crop a search region $x$ in $X_t$ using $b_{t-1}$;
    Extract features $\left\{\varphi^l(x)\right\}_{l=3}^{5}$ for $x$ from our model;
    Get features $\left\{[\varphi^l(x)]_{cls}\right\}_{l=3}^{5}$ of $x$ from $\left\{\varphi^l(x)\right\}_{l=3}^{5}$;
    Get features $\left\{[\varphi^l(x)]_{reg}\right\}_{l=3}^{5}$ of $x$ from $\left\{\varphi^l(x)\right\}_{l=3}^{5}$;
    Get classification features $\left\{P_{cls-feat}\right\}_{l=3}^{5}$ using Eq. (1);
    Get regression features $\left\{P_{reg-feat}\right\}_{l=3}^{5}$ using Eq. (1);
    Obtain classification map $\{P_{cls}\}_{l=3}^{5} \leftarrow \left\{P_{cls-feat}\right\}_{l=3}^{5}$;
    Obtain regression map $\{P_{reg}\}_{l=3}^{5} \leftarrow \left\{P_{reg-feat}\right\}_{l=3}^{5}$;
    Regress $(l, t, r, b)$ via the propose-and-refine module;
    Calculate the final classification map $P_{cls-all}$;
    Calculate the final regression map $P_{reg-all}$;
    Get predicted bounding boxes using Eq. (7);
    Select the optimal bounding box as tracking results $b_t$.
end
```

4. Experiments

In this section, we conduct extensive experiments on six benchmarks (i.e., VOT2018 [28], VOT2019 [29], OTB100 [31], LaSOT [27], UAV123 [32], and NfS [30]) to validate the performance of the proposed PRTracker. Section 4.1 first introduces the implementation details of the proposed PRTracker. Then, we systematically compare it with state-of-the-art trackers in Section 4.2. Finally, Section 4.3 discusses ablation studies on the propose-and-refine module and target mask module.

4.1. Implementation Details

We initialized the backbone network with the parameters pre-trained on ImageNet [52] and freeze the parameters of the first two layers. We trained the proposed network with stochastic gradient descent (SGD) with a minibatch of 28 pairs. We train a total of 20 epochs, using a warmup learning rate of 0.001 to 0.005 in the first 5 epochs and a learning rate exponentially decayed from 0.005 to 0.00005 in the last 15 epochs. In the first 10 epochs, we only trained the propose-and-refine module. In the last 10 epochs, we fine-tuned the backbone network at one-tenth of the current learning rate. Weight decay and momentum were set as 0.0001 and 0.9, respectively. Our PRTracker was implemented under the PyTorch framework. The training was carried out on a workstation with Intel(R) Xeon(R) Silver 4114 2.20 GHz CPU, Nvidia GTX 1080Ti. Our PRTracker runs at 75 FPS.

4.2. Comparison with State-of-the-Art Trackers

VOT2018 [28]. The Visual Object Tracking Challenge 2018 (VOT2018) dataset comprises 60 video sequences with an average length of 356 frames. It evaluates trackers in terms of accuracy (average overlap during successful tracking periods), robustness (failure rate), and EAO (expected average overlap). The EAO is used to rank the trackers based on a combination of the accuracy and robustness metrics. The evaluation on VOT2018 was performed by an official toolkit. Table 3 and Figure 4 show the detailed comparison with nine top-performing trackers, including SiamRPN [20], LADCF [56], ATOM [43], Siam R-CNN [7], SiamRPN++ [9], SiamFC++ [23], DiMP [57], SiamBAN [6], and Ocean [8], on VOT2018. It can be seen that the proposed PRTracker achieved the best EAO, accuracy and robustness. In terms of EAO, the proposed PRTracker was 5.7% higher than the second ranked Ocean [8]. Among the compared trackers, Siam R-CNN [7] achieved pretty high accuracy. However, our PRTracker outperformed it with 1%. It is worth noting that PRTracker achieved the same robustness as DiMP [57] without online updates.

Table 3. Detailed comparisons on VOT2018. The best two results are highlighted in red and blue fonts. DiMP is the ResNet-50 version (DiMP-50), Ocean is the offline Ocean, the same below.

	SiamRPN [20]	LADCF [56]	ATOM [43]	Siam R-CNN [7]	SiamRPN++ [9]	SiamFC++ [23]	DiMP [57]	SiamBAN [6]	Ocean [8]	PRTracker
EAO (↑)	0.384	0.389	0.401	0.408	0.417	0.430	0.441	0.452	0.470	0.497
Accuracy (↑)	0.588	0.503	0.590	0.617	0.604	0.590	0.597	0.597	0.603	0.627
Robustness (↓)	0.276	0.159	0.201	0.220	0.234	0.173	0.150	0.178	0.164	0.150

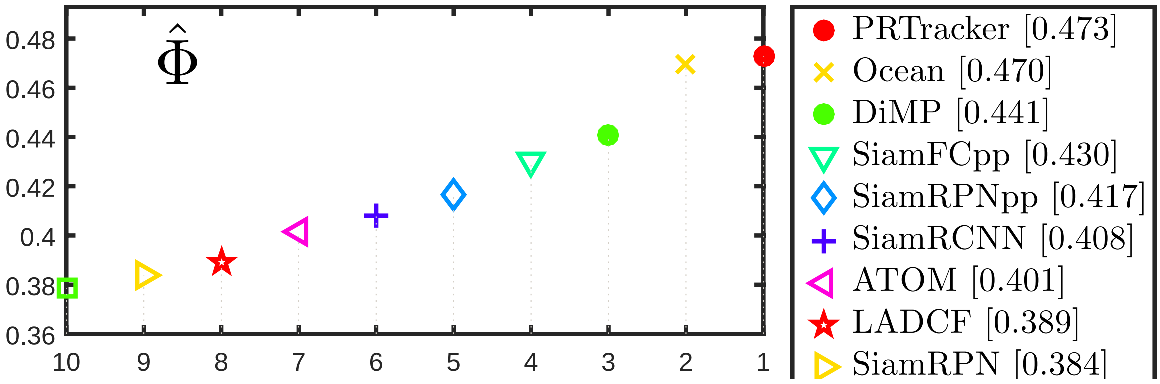


Figure 4. Expected averaged overlap performance on VOT2018. SiamRPNpp is SiamRPN++, SiamFCpp is SiamFC++, SiamRCNN is Siam R-CNN, the same below.

Furthermore, as shown in Figure 5, we provide the qualitative comparison results on four typical sequences, i.e., girls, butterfly, fish2 and motocross1 sequences. The challenges in the four sequences contain similar objects, non-rigid object deformation, background clutters and fast motion, respectively. To be specific, the first row shows that an anchor-free-based tracker (i.e., SiamBAN) and coarse-to-fine-based tracker (i.e., Siam R-CNN) are easily affected by similar interferences. With the use of fixed sampling points of conventional convolution, these trackers cannot accurately localize the objects due to insufficient feature representation. Moreover, the butterfly sequence on the second row with serious object deformation results in poor performance of the other four trackers (i.e., Ocean, DiMP, SiamBAN and Siam R-CNN). In contrast, the proposed PRTracker has a powerful ability to couple this case. We conclude that the propose-and-refine mechanism with an alignment convolution can learn more robust features of deformed objects. As verified in the rest rows, both DiMP with online update and Ocean fail to generate precise object localization results. In conclusion, the above-detailed analyses verify that our PRTracker can learn to propose and refine to effectively improve the convolution sampling position for more accurate feature extraction.

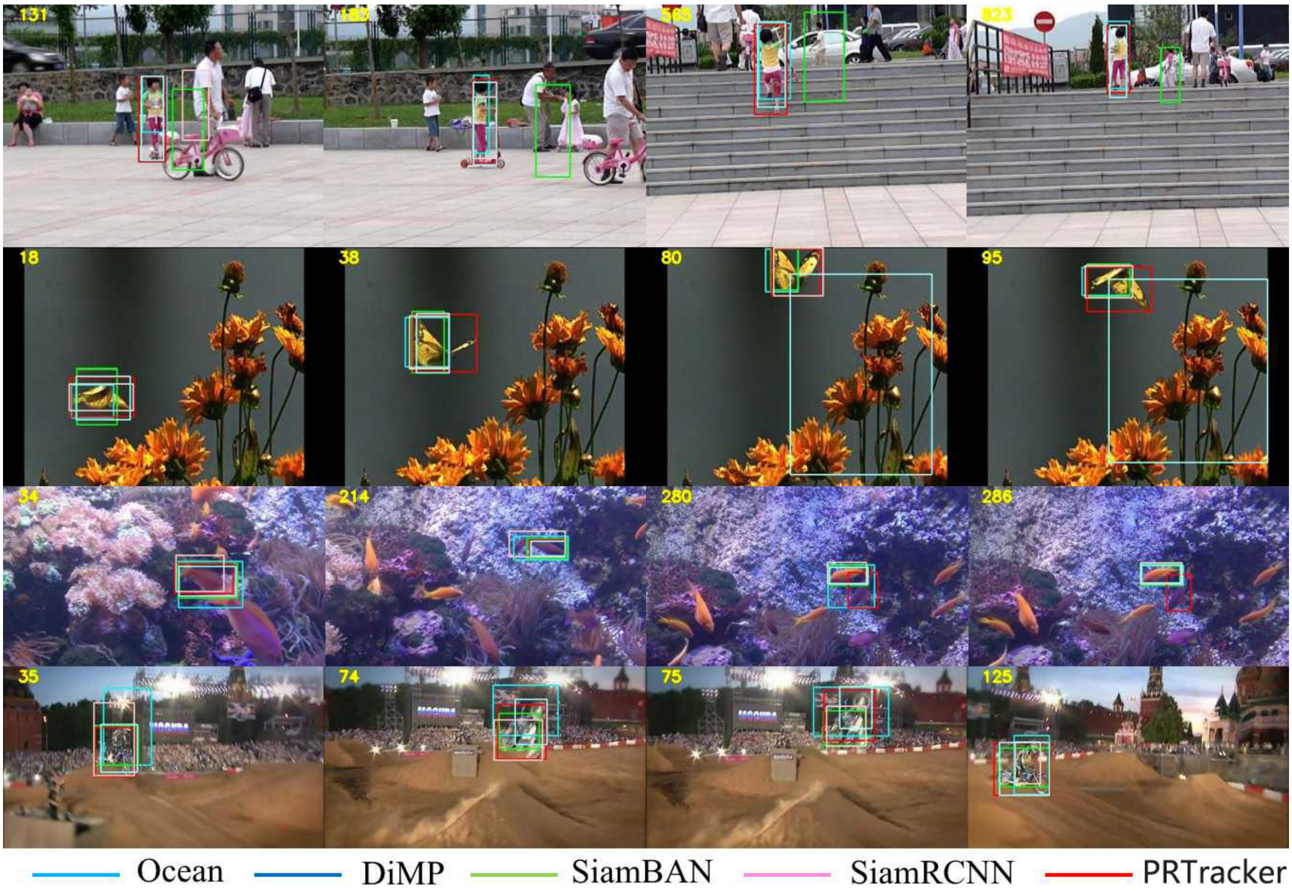


Figure 5. Representative experimental results from the proposed tracker and four state-of-the-art trackers, DiMP [57], SiamBAN [6], SiamRCNN [7] and Ocean [8]. Estimating the target state accurately is challenging due to the existence of similar objects (first row), non-rigid object deformation (second row), background clutters (third row), and fast motion (fourth row). Different from the state-of-the-art trackers, we present a propose-and-refine mechanism for accurate tracking. The proposed tracker obtains an overall performance improvement by further refinement and relatively accurate features.

Comparison of attributes on VOT2018. To clearly show the effectiveness of the proposed PRTracker on various scenes of VOT2018, we study and analyse each attribute in VOT2018. Each frame of all video sequences in VOT2018 is marked with attributes (e.g., camera motion, illumination change, occlusion, size change, or motion change) and the rest are classified as unassigned. We compare the EAO of the major attributes on VOT2018 in Figure 6. It can be clearly seen that our PRTracker achieved a competitive performance in all attributes. It achieved the highest EAO on the attributes of occlusion (e.g., basketball and tiger sequences), and camera motion (e.g., car1 and crossing sequences). Meanwhile, it ranked second on the attributes of motion change (e.g., bag, birds1 and bmx sequences), and size change (e.g., matrix sequence). The excellent performance shows that the proposed PRTracker is able to mitigate inaccurate bounding box regression effects by integrating an alignment convolution into the refinement stage, capturing representative features of the proposals to deal with multiple challenges.

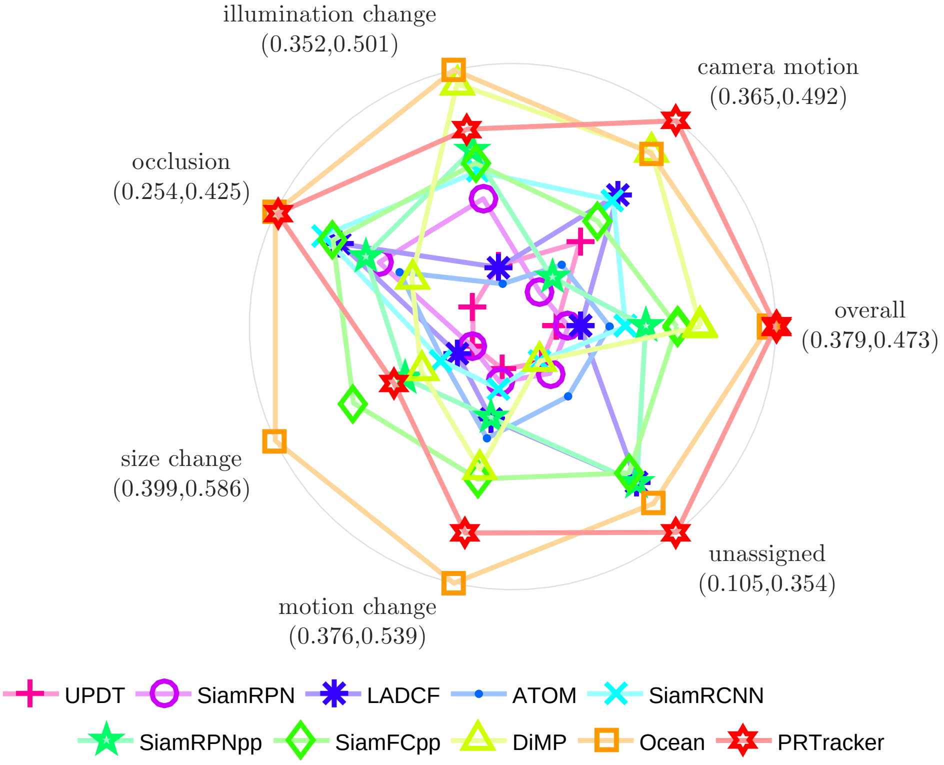


Figure 6. Comparison of EAO on VOT2018 for the following visual attributes: camera motion, illumination change, occlusion, size change and motion change. Frames that do not correspond to any of the five attributes are marked as unassigned. The values in parentheses indicate the EAO range of each attribute and overall of the trackers.

VOT2019 [29]. The Visual Object Tracking challenge 2019 (VOT2019) dataset is obtained by replacing 20% of VOT2018 with carefully selected sequences from the GOT10k [55]. The same metrics and tools from VOT2018 are used for performance evaluation. Table 4 and Figure 7 show the EAO, robustness and accuracy of the proposed PRTracker and the nine top-performing trackers, respectively. Compared with these trackers, the proposed PRTracker achieved the best overall performance with the highest accuracy and lowest robustness. Furthermore, the proposed PRTracker outperformed the other coarse-to-fine-based trackers (e.g., SPM [10] and SiamMask [35]). These comparison results clearly verify the advantage of our designed propose-and-refine module. Moreover, the performance of the proposed PRTracker exceeds DiMP [57] with online updates. These results show that our PRTracker can effectively use an alignment convolution in the refine stage to obtain more accurate and robust features for target localization.

Table 4. Detailed comparisons on VOT2019 experiments. DiMP is real-time version, as reported in [29]. The best two results are highlighted in red and blue fonts. The uparrow signs mean that the higher the score, the better. The downarrow signs mean that the lower the score, the better.

	SPM [10]	**SiamRPN++** [9]	**SiamMask** [35]	**ARTCS** [29]	**SiamDW_ST** [34]	**DCFST** [29]	**DiMP** [57]	**SiamBAN** [6]	**Ocean** [8]	**PRTracker**
EAO (↑)	0.275	0.285	0.287	0.287	0.299	0.317	0.321	0.327	0.329	0.352
Accuracy (↑)	0.577	0.599	0.594	0.602	0.600	0.585	0.582	0.602	0.595	0.634
Robustness (↓)	0.507	0.482	0.461	0.482	0.467	0.376	0.371	0.396	0.376	0.341

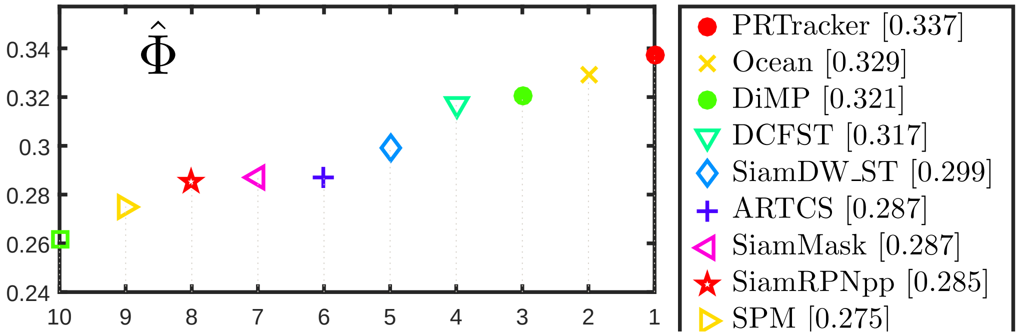
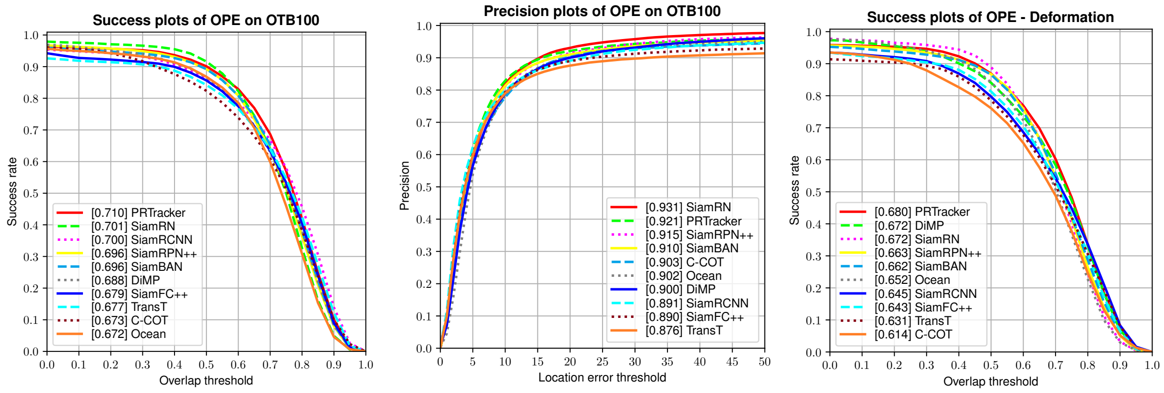


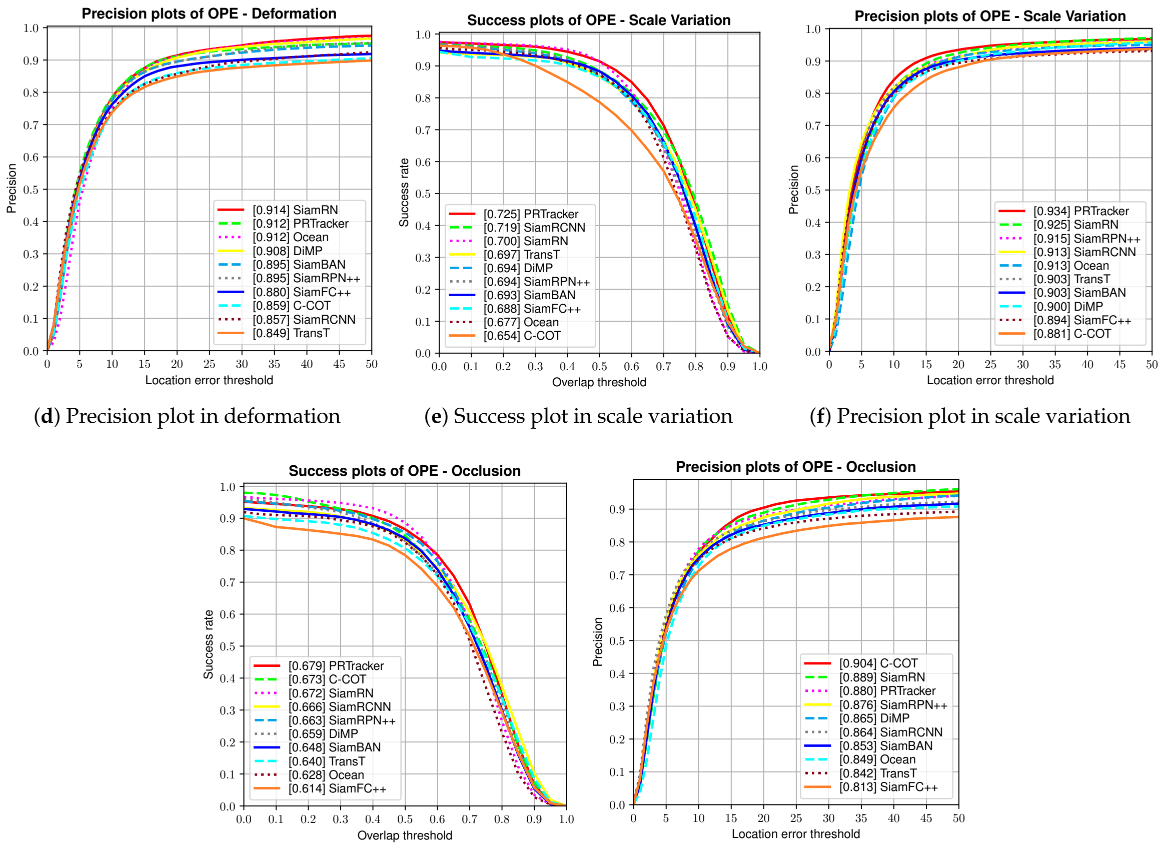
Figure 7. Expected averaged overlap performance on VOT2019.

OTB100 [31]. The two standard evaluation metrics on OTB100 are success rate and precision. For each frame of a video sequence, we computed the *IoU* between the predicted bounding boxes and the ground truth, and the distance of their central locations. A success plot can be generated by evaluating the success rate under different *IoU* thresholds. By convention, we report the area under the curve (AUC) of the success plot. The precision plots can be obtained via a similar way, usually reporting the accuracy at a 20 pixel threshold. The proposed tracker was compared with state-of-the-art trackers including Siam R-CNN [7], SiamRPN++ [9], SiamBAN [6], SiamRN [58], DiMP [57], SiamFC++ [23], Ocean [8], C-COT [38], and TransT [59]. Figure 8a,b illustrates the success and precision plots. Our tracker obtained an AUC score of 0.710 and a precision score of 0.921, higher than the previous best results from SiamRN. Meanwhile, compared to the transformer-based tracker (i.e., TransT [59]), the proposed PRTracker improves by 3.3 and 5.5% in the success and precision metrics, respectively. In addition, we further compared each attribute on the OTB100 in Figure 8c–h. Overall, the proposed PRTracker achieved good results in various attributes, especially in the challenges of deformation and scale variations. In these cases, the bounding boxes of the compared trackers were less accurate than the PRTracker. This is because the significantly scale and aspect ratio variations require the trackers to have a robust and precise feature extraction ability. Notably, this is consistent with our key motivation that precise and robust feature extraction plays a vital role in accurate tracking. The design of the proposed PRTracker can generate accurate features through adjusting the sampling points using the alignment convolution with the supervision information. Thus, our PRTacker achieved a promising performance.



(**a**) Success plot (**b**) Precision plot (**c**) Success plot in deformation

Figure 8. *Cont.*



(**d**) Precision plot in deformation (**e**) Success plot in scale variation (**f**) Precision plot in scale variation

(**g**) Success plot in occlusion (**h**) Precision plot in occlusion

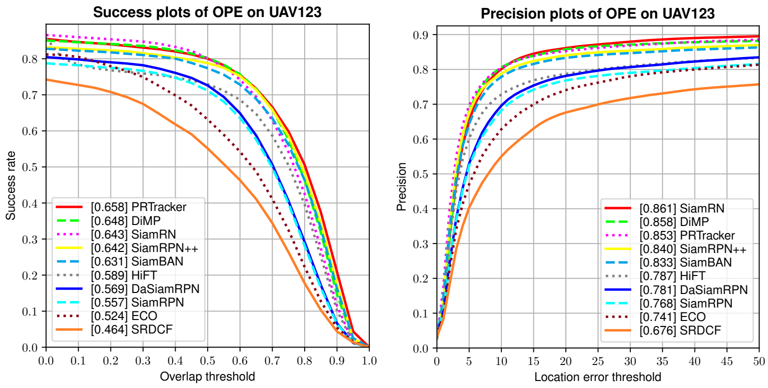
Figure 8. The success and precision plots on OTB100.

NfS [30]. The Need for Speed (NfS) dataset consists of 100 videos (380K frames) captured from real-world scenes with higher frame rate cameras. All frames are annotated with axis-aligned bounding boxes, and all sequences are manually labelled with nine visual attributes such as occlusion, fast motion, background clutter, etc. We report results on the 30 FPS version of the dataset. As shown in Table 5, our PRTracker achieved competitive tracking performance and ranked second. DiMP outperformed our PRTracker by 1.7%. We surmise that one of the underlying reasons is that the target appearances in higher frame rate videos significantly change. The trackers (e.g., DiMP) using online updating strategies can effectively handle significant target appearance variations. Without using an online updating strategy, the proposed PRTracker still obtained a promising performance.

Table 5. Comparison with state-of-the-art trackers on the NfS dataset in terms of AUC. The best two results are highlighted in red and blue fonts. The uparrow signs mean that the higher the score, the better.

	MDNet [60]	**ECO** [33]	**C-COT** [38]	**UPDT** [61]	**ATOM** [43]	**SiamBAN** [6]	**DiMP** [57]	**PRTracker**
AUC (↑)	0.422	0.466	0.488	0.537	0.584	0.594	0.620	0.603

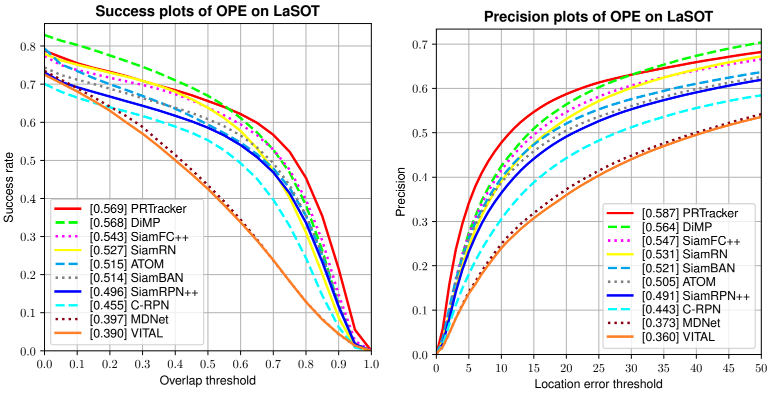
UAV123 [32]. Different from other tracking datasets, such as OTB100, VOT2018, VOT2019 and NfS, UAV123 is collected from low-altitude UAVs and contains a total of 123 video sequences. The length of the videos in UAV123 is more than 110K frames. We compared the proposed PRTracker with nine state-of-art trackers, including DiMP [57], HiFT [62], SiamRPN++ [9], SiamBAN [6], DaSiamRPN [21], SiamRPN [20], ECO [33], SRDCF [63], SiamRN[58]. Figure 9 shows the comparison results. Although a slightly lower precision score was achieved, the proposed PRTracker obtained a better success score against DiMP, which is equipped with a updating strategy. Meanwhile, compared to the transformer-based tracker (i.e., HiFT [62]), the proposed PRTracker improved by 6.9 and 6.6% in the success and precision metrics, respectively. The above observations demonstrate the effectiveness of the proposed PRTracker on UAV123. This is because our PRTracker can effectively extract accurate and robust features via the propose-and-refine module driven by an alignment convolution.



(**a**) Success plot (**b**) Precision plot

Figure 9. The success and precision plots on UAV123.

LaSOT [27]. Compared with short-term tracking datasets (e.g., OTB100 [31], VOT2018 [28], VOT2019 [29], NfS [30] and UAV123 [32]), LaSOT has longer sequences with an average sequence length of more than 2500 frames. We compared our PRTracker on the testing set consisting of 280 videos with trackers including DiMP [57], SiamFC++ [23], SiamRN [58], ATOM [43], SiamBAN [6], SiamRPN++ [9], C-RPN [64], MDNet [60] and VITAL [65]. The success and precision plots are shown in Figure 10. Compared with SiamFC++ based on a Siamese network, the proposed PRTracker outperformed it by 2.6 and 4% in the AUC and precision, respectively. Moreover, compared with DiMP that uses an effective model prediction and updating strategy, we observed improvements of +2.3% in the precision. Meanwhile, compared to the SiamRN [58], the proposed PRTracker improved by 4.2 and 5.6% in the success and precision metrics, respectively. The above experimental results further verify the importance of accurate and robust feature extraction for the tracking performance. Overall, our propose-and-refine module can obtain more powerful features via an alignment convolution. As a result, the proposed PRTracker is applicable to more challenges in longer sequences.



(**a**) Success plot (**b**) Precision plot

Figure 10. The success and precision plots on LaSOT.

4.3. Ablation Study

To validate the effectiveness of each component in the proposed PRTracker, we explored the roles of the propose-and-refine module, alignment convolution and target mask. Table 6 shows our ablation analysis on OTB100 [31] and LaSOT [27].

Discussion on our propose-and-refine module and alignment convolution. Firstly, we verified the proposed propose-and-refine module to effectively extract accurate and robust features for object tracking and localization. Then, we show how the three convolution methods (i.e., the conventional, deformable and alignment convolution) affect our PRTracer on the OTB100 [31] and LaSOT [27] datasets.

As shown in Table 6, we evaluated four degraded trackers of our PRTracker by removing each component and validating the performance on OTB100 and LaSOT. The four degraded trackers are denoted as T1, T2, T3, and T4, respectively. As shown in the first row of Table 6, a baseline tracker (denoted as T1) consists of a backbone, regression and classification sub-network. Both regression and classification sub-networks firstly use three 3 × 3 convolution layers to generate corresponding feature maps. Then, the regression and classification feature maps use a 3 × 3 convolution layer and a 1 × 1 convolution layer to predict the bounding boxes and classification scores, respectively. The AUCs of T1 on OTB100 and LaSOT were 0.683 and 0.512, respectively. In the second row of Table 6, we constructed a tracker (denoted as T2) by adding a 3 × 3 convolution layer and a 1 × 1 convolution layer after the regression sub-network to fine-tune the initial bounding boxes. We observed improvements of +0.7 and +1.1% on OTB100 and LaSOT, respectively. These results validate the importance of the refinement strategy. In the third row of Table 6, we constructed a tracker (denoted as T3) by using a 3 × 3 deformable convolution substituting for the 3 × 3 conventional convolution of T2. T3 obtained 0.4 and 0.7% improvements on OTB100 and LaSOT, respectively, than T2. The reason for this is that the deformable convolution can augment the spatial sampling locations in the modules with additional offsets and learn the offsets from the target tasks. In contrast, the conventional convolutions are limited to model geometric transformations due to the fixed geometric structures. However, one of the limitations of the deformable convolutions is that they obtain sampling positions without direct supervision. To address this issue, the alignment convolutions obtain sampling positions based on a bounding box, and thus can capture more geometric and context information of the bounding box (detailed in Section 3.3). As shown in the fourth row of Table 6, we constructed a tracker (denoted as T4) using a 3 × 3 alignment convolution layer and a 1 × 1 convolution layer to classify and refine the bounding boxes, respectively. T4 exhibited improvements of +1.2 and +3.3% on OTB100 and LaSOT, respectively.

Table 6. Quantitative comparison results of the proposed PRTracker with different convolution operations and target masks on OTB100 and LaSOT. CC, DC, AC represent conventional, deformable and alignment convolution, respectively. The four degraded trackers of our PRTracker are denoted as T1, T2, T3, and T4, respectively.

	CC	**DC**	**AC**	**Target Mask**	**OTB100 AUC**	**LaSOT AUC**
T1					0.683	0.512
T2	✓				0.690	0.523
T3		✓			0.694	0.530
T4			✓		0.702	0.556
PRTracker			✓	✓	0.710	0.569

Discussion of the proposed target mask. In the fifth row of Table 6, we constructed the proposed PRTracker by introducing a target mask (detailed in Section 3.4) into T4. The goal of our target mask was to make full use of the template information to improve the features' discriminative power between a target object and its nearby background. As a benefit, the final AUCs of our PRTracker on OTB100 and LaSOT were 0.710 and 0.569, respectively. The results are consistent with our idea that the target mask enables our PRTracker to pay more attention to a target and distinguish between the target and background.

Based on the ablation studies, our propose-and-refine module driven by an alignment convolution significantly improves the performance of the proposed PRTracker. The designed alignment convolution with the supervision information is able to extract more reliable features for accurate and robust object localization. Moreover, the target mask, introduced into our PRTracker without additional computational burden, can further improve the tracking performance.

5. Conclusions

In this paper, we proposed a PRTracker that utilizes a propose-and-refine mechanism driven by an alignment convolution to classify and refine proposals. The proposed PRTracker combines anchor-free style proposals at the coarse level, and alignment convolution-driven refinement at the fine level. As a benefit, the proposed PRTracker can effectively address how to choose appropriate convolutional sampling points for accurate and robust feature extraction. Moreover, we designed a simple yet robust target mask to naturally use the initial state of a target to enhance the tracking performance. The encouraging results on six tracking benchmarks demonstrate the accuracy and robustness of our PRTracker with 75 FPS.

In the future work, we aim to explore state-of-the-art detection strategies to further improve the tracking results, e.g., such as YOLO versions and transformers.

Author Contributions: Z.M. discovered that inaccurate feature representation degrades the tracking performance, highlighting that current conventional or deformable convolutions extract features from the sampling positions that may be far away from a target region, potentially introducing background noise. Thus, he proposed the main idea of this paper. Furthermore, he implemented the experiments of alignment convolution. He proposed the use of the target mask and performed experiments to confirm its validity. Furthermore, he wrote the article, including describing the methods, drawing the figures, and designing the tables. Z.L. sorted the experimental results and drew the visualization results for further performance verification. He also participated in discussing the proposed framework, performing experiments and the final writing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the Guangxi Talent Highland Project of Big Data Intelligence and Application.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

AlignConv	Alignment convolution
AUC	Area under the curve
DCF	Discriminative correlation filter
DW-Corr	Depth-wise cross-correlation operation
EAO	Expected average overlap
NfS	Need for speed
PRTracker	Propose-and-refine tracker

RPN	Region proposal network
SGD	Stochastic gradient descent
FPS	Frame per second
RoI	Region of interest
VOT2018	Visual Object Tracking Challenge 2018
VOT2019	Visual Object Tracking challenge 2019

References

1. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R.; Tang, Z.; Li, X. SiamBAN: Target-aware tracking with siamese box adaptive network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5158–5173. [CrossRef] [PubMed]
2. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3D Convolutional Neural Network. In Proceedings of the CVPR 2016, Las Vegas, NV, USA, 26 June–1 July 2016.
3. Lee, K.H.; Hwang, J.N. On-Road Pedestrian Tracking Across Multiple Driving Recorders. *IEEE Trans. Multimed.* **2015**, *17*, 1. [CrossRef]
4. Tang, S.; Andriluka, M.; Andres, B.; Schiele, B. Multiple People Tracking by Lifted Multicut and Person Re-identification. In Proceedings of the CVPR 2017, Honolulu, HI, USA, 22–25 July 2017.
5. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision Workshops, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
6. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, virtual, 14–19 June 2020; pp. 6668–6677.
7. Voigtlaender, P.; Luiten, J.; Torr, P.H.; Leibe, B. Siam r-cnn: Visual tracking by re-detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, virtual, 14–19 June 2020; pp. 6578–6588.
8. 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.
9. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291.
10. Wang, G.; Luo, C.; Xiong, Z.; Zeng, W. SPM-Tracker: Series-parallel matching for real-time visual object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3643–3652.
11. Mu, Z.A.; Hui, Z.; Jing, Z.; Li, Z. Multi-level prediction Siamese network for real-time UAV visual tracking. *Image Vis. Comput.* **2020**, *103*, 104002. .
12. Wu, Y.; Liu, Z.; Zhou, X.; Ye, L.; Wang, Y. ATCC: Accurate tracking by criss-cross location attention. *Image Vis. Comput.* **2021**, *111*, 104188. [CrossRef]
13. Zheng, Y.; Zhong, B.; Liang, Q.; Tang, Z.; Ji, R.; Li, X. Leveraging Local and Global Cues for Visual Tracking via Parallel Interaction Network. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1671–1683. [CrossRef]
14. Ma, J.; Lan, X.; Zhong, B.; Li, G.; Tang, Z.; Li, X.; Ji, R. Robust Tracking via Uncertainty-aware Semantic Consistency. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1740–1751. [CrossRef]
15. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
16. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the ICCV, IEEE Computer Society, Venice, Italy, 22–29 October 2017; pp. 764–773.
17. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable Siamese Attention Networks for Visual Object Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 6727–6736. [CrossRef]
18. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
19. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning attentions: residual attentional siamese network for high performance online visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 4854–4863.
20. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 8971–8980.
21. 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, Munich, Germany, 8–14 September 2018; pp. 101–117.
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 10–17 October 2015; pp. 91–99.
23. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In Proceedings of the AAAI, New York, NY, USA, 7–12 February 2020; pp. 12549–12556.

24. 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 Conference on Computer Vision and Pattern Recognition, virtual, 14–19 June 2020; pp. 6269–6277.
25. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 22–25 July 2017; pp. 2961–2969.
26. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of localization confidence for accurate object detection. In Proceedings of the European Conference on Computer Vision, Salt Lake City, UT, USA, 19–21 June 2018; pp. 784–799.
27. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5374–5383.
28. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. The sixth visual object tracking VOT2018 challenge results. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3-53.
29. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pflugfelder, R.; Kamarainen, J.K.; Cehovin Zajc, L.; Drbohlav, O.; Lukezic, A.; Berg, A.; et al. The seventh visual object tracking VOT2019 challenge results. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1–36.
30. Kiani Galoogahi, H.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. Need for speed: A benchmark for higher frame rate object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1125–1134.
31. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
32. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 445–461.
33. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 6638–6646.
34. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
35. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338.
36. Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.H. Robust Visual Tracking via Convolutional Networks Without Training. *IEEE Trans. Image Process.* **2016**, *25*, 1779–1792. [CrossRef] [PubMed]
37. Zheng, Y.; Liu, X.; Cheng, X.; Zhang, K.; Wu, Y.; Chen, S. Multi-Task Deep Dual Correlation Filters for Visual Tracking. *IEEE Trans. Image Process.* **2020**, *29*, 9614–9626. [CrossRef] [PubMed]
38. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 472–488.
39. 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.
40. Xie, S.; Girshick, R.B.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]
41. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
42. Yan, B.; Zhang, X.; Wang, D.; Lu, H.; Yang, X. Alpha-Refine: Boosting Tracking Performance by Precise Bounding Box Estimation. *arXiv* **2020**, arXiv:2012.06815.
43. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate tracking by overlap maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4660–4669.
44. Gomaa, A.; Abdelwahab, M.M.; Abo-Zahhad, M. Efficient vehicle detection and tracking strategy in aerial videos by employing morphological operations and feature points motion analysis. *Multim. Tools Appl.* **2020**, *79*, 26023–26043. [CrossRef]
45. Gomaa, A.; Minematsu, T.; Abdelwahab, M.M.; Abo-Zahhad, M.; Taniguchi, R. Faster CNN-based vehicle detection and counting strategy for fixed camera scenes. *Multim. Tools Appl.* **2022**, *81*, 25443–25471. [CrossRef]
46. Chang, Y.; Tu, Z.; Xie, W.; Luo, B.; Zhang, S.; Sui, H.; Yuan, J. Video anomaly detection with spatio-temporal dissociation. *Pattern Recognit.* **2022**, *122*, 108213. [CrossRef]
47. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
48. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]

49. Law, H.; Deng, J. CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 734–750.
50. Jang, H.D.; Woo, S.; Benz, P.; Park, J.; Kweon, I.S. Propose-and-attend single shot detector. In Proceedings of the The IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 815–824.
51. Zhang, H.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Cascade retinanet: Maintaining consistency for single-stage object detection. *arXiv* **2019**, arXiv:1907.06881.
52. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
53. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 5296–5305.
54. 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 Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
55. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A large high-diversity benchmark for Ggeneric object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *43*, 1562–1577. [CrossRef] [PubMed]
56. Xu, T.; Feng, Z.H.; Wu, X.J.; Kittler, J. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE Trans. Image Process.* **2019**, *28*, 5596–5609. [CrossRef] [PubMed]
57. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.
58. Cheng, S.; Zhong, B.; Li, G.; Liu, X.; Tang, Z.; Li, X.; Wang, J. Learning To Filter: Siamese Relation Network for Robust Tracking. In Proceedings of the CVPR. Computer Vision Foundation/IEEE, virtual, 19–25 June 2021; pp. 4421–4431.
59. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the CVPR. Computer Vision Foundation/IEEE, virtual, 19–25 June 2021; pp. 8126–8135.
60. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4293–4302.
61. Bhat, G.; Johnander, J.; Danelljan, M.; Shahbaz Khan, F.; Felsberg, M. Unveiling the power of deep tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 483–498.
62. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. HiFT: Hierarchical Feature Transformer for Aerial Tracking. In Proceedings of the ICCV. IEEE, Montreal, QC, Canada, 10–17 October 2021; pp. 15437–15446.
63. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
64. Fan, H.; Ling, H. Siamese cascaded region proposal networks for real-time visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7952–7961.
65. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.; Yang, M.H. VITAL: Visual tracking via adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 8990–8999.

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.