Part-Aware Framework for Robust Object Tracking

Shengjie Li, Shuai Zhao, *Member, IEEE*, Bo Cheng, *Member, IEEE*, and Junliang Chen

***Abstract*— The local parts of the target are vitally important for robust object tracking. Nevertheless, existing excellent context regression methods involving siamese networks and discrimination correlation filters mostly represent the target appearance from the holistic model, showing high sensitivity in scenarios with partial occlusion and drastic appearance changes. In this paper, we address this issue by proposing a novel part-aware framework based on context regression, which simultaneously considers the global and local parts of the target and fully exploits their relationship to be collaboratively aware of the target state online. To this end, the spatial-temporal measure among context regressors corresponding to multiple parts is designed to evaluate the tracking quality of each part regressor by solving the imbalance among global and local parts. The coarse target locations provided by part regressors are further aggregated by treating their measures as weights to refine the final target location. Furthermore, the divergence of multiple part regressors in each frame reveals the interference degree of background noise, which is quantified to control the proposed combination window functions in part regressors to adaptively filter redundant noise. Besides, the spatial-temporal information among part regressors is also leveraged to assist in accurately estimating the target scale. Extensive evaluations demonstrate that the proposed framework help many context regression trackers achieve performance improvements and perform favorably against state-of-the-art methods on the popular benchmarks: OTB, TC128, UAV, UAVDT, VOT, TrackingNet, GOT-10k, LaSOT.**

***Index Terms*— Object tracking, siamese network, discrimination correlation filter, global and local parts.**

I. Introduction

VISUAL object tracking is one of the fundamental research topics in image processing with a wide range of applications such as action recognition, video surveillance and human-computer interaction [1]. Although great progress has been made in recent years [2], [3], [4], [5], it is still a challenging problem to develop a robust tracking approach that can estimate the target state in scenarios with partial occlusion, out-of-view, scale changes, deformation and Wu et al. [6], [7], [8].

In recent years, both siamese network [10], [11], [12], [13] and discriminative correlation filter (DCF) [14], [15], [16], [17] based methods adopt the context regression strategy to achieve superior performance on the large-scale benchmarks [9], [18], [19], [20]. Early DCF trackers usually regress all circular-shifted samples about input features into soft labels from the Gaussian distribution and convert the correlation of the target and surrounding context in spatial domains to the element-wise product in Fourier domains. On this basis, some works [16], [21] have recently applied deep learning frameworks to solve the above regression problem, which avoids the boundary effect of early DCF trackers. Borrowing from the idea of correlation in DCF, most siamese trackers adopt the cross-correlation operation between temporal context to regress the outputs of two-fold fully convolutional networks to the Gaussian distribution labels. The recent advances [11], [13] show that the accurate target scale estimation in siamese trackers can be efficiently solved by replacing the bruteforce multi-scale search with the bounding box regression. Although achieved the appealing tracking performance both in accuracy and robustness, most siamese and DCF trackers only explore the holistic target representation by the context regressors and ignore the detailed local target representation, thus leading to the room for improvement due to their high sensitivity to challenging scenarios where partial occlusion and drastic variations of the target appearance happen.

Compared to representing the target appearance with a global model, there are some attempts [22], [23], [24] to deal with these highly sensitive issues by applying part-based strategies to the efficient context regressors from two aspects. The first one is to explore the power of the local features of the target by offline training [25], [26], thereby realizing online real-time tracking. The other one without offline training is to partition the target region into multiple local parts online and employ the context regressors for each part to keep track of the target in parallel [27], [28]. As a consequence, these trackers usually have the advantage of being robust against partial occlusions, since the reliable cues can still be provided by the remaining visible parts from the target for tracking. Whilst these part-based approaches mitigate the difficulties in handling non-rigid appearance variations, a potential limitation is that most of them fail to make full use of the relationship

Manuscript received 6 June 2022; revised 9 November 2022; accepted 7 December 2022. Date of publication 5 January 2023; date of current version 10 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20468, Grant 61921003, Grant 61972043, and Grant 52071312; in part by the Beijing Nova Program of Science and Technology under Grant Z191100001119031; in part by the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS202111; in part by the China Postdoctoral Science Foundation under Grant 2021M700516; and in part by the Open Program of Zhejiang Laboratory under Grant 2021PD0AB02. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giuseppe Valenzise. *(Corresponding authors: Shuai Zhao; Bo Cheng.)*

Shengjie Li, Bo Cheng, and Junliang Chen are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lishengjie@bupt.edu.cn; chengbo@bupt.edu.cn; chjl@bupt.edu.cn).

Shuai Zhao is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China (e-mail: zhaoshuaiby@bupt.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2022.3232941, provided by the authors.

Digital Object Identifier 10.1109/TIP.2022.3232941

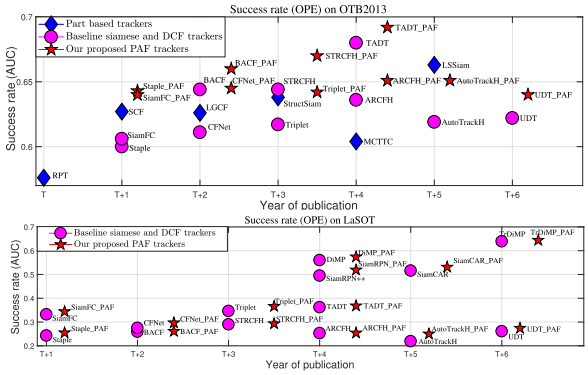

Fig. 1. The success rate (AUC) obtained by the part-based (blue color and published since T = 2015 year), baseline context regression (*e.g.*, siamese or DCF denoted by magenta) and proposed (represented by red color) trackers on OTB2013 [6] and LaSOT [9]. Our part-aware framework (PAF) can help many baseline context regression trackers obtain performance improvements and outperform other part-based trackers as well as state-of-the-art methods, which shows the generalization and great potential of our framework.

among the global and local parts in both spatial and temporal domains, which helps to alleviate the problem of model drift away in the situation of heavy occlusion or deformation.

To address the above issues mentioned, we in this paper explore to design a generic part-aware framework (PAF) for robust object tracking, aiming to improve the robustness of both siamese and DCF trackers by introducing the global and local parts to these context regressors as shown in Figure 1. In addition to each part corresponding with a context regressor (*e.g.*, siamese or DCF), which follows the idea of part-based trackers without offline training, the key innovation is that we focus on fully exploiting the relationship of outputs among global and local part regressors to being collaboratively aware of the location, noise interference and scale of the target. Specifically, the tracking output quality of each part regressor is evaluated from two aspects: spatial measure and temporal measure. The former pays attention to carefully describing the relationship of the bounding box outputs of part regressors in the spatial domain, while the latter focuses on precisely representing the relationship of the response outputs of each part regressor in the temporal domain. Subsequently, the coarse predictions provided by part regressors and these spatial-temporal measures as the weights are adaptively aggregated together to estimate the final target location. To further improve the robustness of each part regressor, we take into consideration the combination window functions in these regressors based on their tracking quality to adaptively filter the background noise during tracking. In addition, the spatial-temporal output information of part regressors is also considered to provide the auxiliary target scale estimation, which is different from the aforementioned part-based trackers. Experimental evaluations and analyses on the large-scale benchmarks demonstrate the effectiveness and efficiency of the proposed framework. The main contributions of this work can be summarized as follows:

- We propose a generic part-aware framework for context regression tracking by fully exploiting global and local parts of the target and the output relationship of part regressors to be collaboratively aware of the target state.
- We develop a novel spatial-temporal measure among multiple parts of the target to evaluate the tracking quality of each part regressor by solving the output imbalance problem among the global and local part regressors.
- We design a combination window function including the constant and variation windows for each part regressor by considering the output divergence of all part regressors to adaptively suppress the redundant noise.

Our framework can be applied to many siamese and DCF trackers. And numerous experiments on the OTB [6], UAV [8], TC128 [7], UAVDT [18], LaSOT [9], TrackingNet [20], VOT [29], [30] and GOT-10k [19] benchmarks show our variants achieve superior tracking performance than baselines, as well as the comparable accuracy with the state-of-the-art trackers.

II. Related Work

Visual object tracking is one of the most popular topics in computer vision with extensive surveys over the past decades [1]. In this section, we briefly review two categories of tracking methods closely related to our work: context regression tracking and part-based tracking.

A. Context Regression Tracking

Benefiting from the context regression scheme, both siamese and DCF methods have successfully been applied to visual tracking and achieve the advantage of being computationally efficient and accurate. On the one hand, siamese trackers have gained significant popularity in recent years, which handle the tracking task by exploiting the temporal context response with two-fold fully convolutional networks [2], [31]. By matching between searching areas of new frames and the initial template, Bertinetto et al. [10] adopt the context regression scheme to cross-correlate the siamese network outputs at the running speed of about 80 frames per second (FPS). Along with the introduction of adaptive update [32], triplet loss [33], graph representation [34], [35], region proposal network [12], [36], [37], feature enhancement [38], [39], [40], unsupervised learning [41], meta-learning [42], segmentation mask [13], [43], anchor-free [44], [45], [46] and network architecture [11], [47], [48], [49], siamese trackers have obtained superior performance in real-time.

On the other hand, the DCF methods regress the circular-shifted samples into Gaussian distribution labels and convert the correlation in spatial domains to Fourier domains. A series of strategies have been developed to optimize DCF trackers, such as kernelized computing [14], feature integration [50], boundary effect [51], [52], [53], ensemble scheme [4], [54] and particle filter [55]. The recent advances show that there are some attempts [15], [17], [56], [57] to solve the ridge regression in the convolutional neural network (CNN) frameworks, which prevents the stubborn boundary effect. Specifically, a discriminative CNN kernel is trained to convolve with the context search area for generating a Gaussian-like response [21], [58], [59]. Subsequently, Danelljan et al. [15] propose the Conjugate Gradient and

Gauss-Newton algorithm for speeding up the kernel training process. To enhance the discriminative capability of the learned CNN kernel, Danelljan et al. [16] fully exploit both target and surrounding background information in an end-to-end architecture, which is further improved by the probabilistic regression [60]. Until very recently, Wang et al. [61] introduce the transformer architecture into the DCF methods based on CNN and achieve state-of-the-art results.

Despite achieving promising results in both accuracy and efficiency, most siamese and DCF trackers usually train a single regressor based on the holistic representation instead of local parts of the target, which leaves the room for improvement especially when the target encounters partial occlusions and drastic appearance variations. Different from them, our framework considers not only the global but also local parts of the target to use multiple regressors to adequately explore the relationship of these parts, of which the outputs are evaluated from the spatial-temporal domains and further aggregated to be collaboratively aware of the target state online.

B. Part-Based Tracking

Structural representation as an effective approach to enhancing robustness has been studied actively in the tracking community. Typically, a keypoint-based structure is adopted to represent the target for tracking. Nebehay et al. [24] propose a hierarchical clustering algorithm for keypoint-based tracking, which distinguishes between inlier and outlier keypoints and improves on state-of-the-art tracking results. Mazzeo et al. [28] exhibit very encouraging performance for keypoint-based tracking by dense SIFT descriptors and the nearest neighbor learning algorithm with template/context matching. Another family of structure representation is part-based methods [22], [26], [27]. Early part-based trackers divided the target appearance into multiple parts online and robustly locate the target by the remaining visible parts in presence of partial occlusion and drastic appearance variations. Liu et al. [62] propose to apply multiple correlation filters on the local parts of the target and explore the adaptive weight method to fuse these parts. Li et al. [63] propose to exploit image patches to model the target appearance and adopt the trackability and motion similarity to search the target. To preserve the target space structure, Liu et al. [64] propose the structure correlation filter by considering the motion model of local parts. On the basis of the structure correlation filter, Fan et al. [65] further consider the global part of the target and the temporal consistency among all parts to obtain promising performance. Burceanu et al. [66] employ a society of tracking parts and co-occurrences constraints to design a dual-pathway network and achieve state-of-the-art performance and robustness. Recently with the introduction of CNN offline training, Zhang et al. [23] propose to explore the local patterns of the target and their spatial relationships with a fast structured siamese network. Zhou et al. [25] present a saliency-associated object tracker by dealing with the discriminative local saliencies and associating saliencies to achieve the global solution and promising performance.

Although appealing results are achieved by the part-based strategy, there still exist some potential limitations: (1) most context regressors applied on the target parts are just regarded as independent, their methods fusing the coarse predictions of regressors are relatively simple and straightforward; (2) few of them jointly explore the spatial-temporal relationship of output information among global and local part regressors to collaboratively estimate the target state; (3) almost all ignore the fact that part regressors use a fixed cosine window to fight against the dynamic background noise interference, which easily causes incorrect location predictions.

Different from the above methods: (1) our framework based on the global and local parts of the target is generic to the context regression scheme, which makes full use of the spatial-temporal relationship of outputs among part regressors. (2) after checking the spatial-temporal measures of part regressor outputs carefully, our framework can effectively refine the tracking results by adaptively aggregating the coarse predictions of part regressors; (3) by considering the output divergence of multiple part regressors, we present combination window functions in these regressors to filter the redundant noise interference (*e.g.*, occlusion or deformation) effectively.

III. Part-Aware Framework

In this work, we aim to introduce the global and local parts of the target to improve the robustness of context regression trackers by a novel part-aware framework, where the spatial-temporal relationship of outputs among part regressors are fully exploited to being collaboratively aware of the target state. Figure 2 illustrates the online tracking process of the overall framework. We can see that the target appearance is first decomposed to one global and several local parts of which each is associated with a context regressor, and the generated outputs of these regressors are denoted by the response maps determining the bounding box locations of parts. Then, the tracking quality of each part regressor is evaluated by the spatial-temporal measure of its response outputs, which is further treated as the weight to adaptively aggregate the coarse target predictions of part regressors together for refining the final target location. Next, the combination window functions including the constant and variable windows are constructed in each part regressor to adaptively filter redundant noise, of which the degree is described by the evaluated spatial-temporal measure and response outputs of part regressors. Finally, the auxiliary scale estimation based on spatial-temporal output information of multiple parts is also considered in final results, which are used as feedback to all part regressors regarding the strictly restricted arrangement of these parts in the next frames. The detailed descriptions are introduced as follows.

A. Global and Local Part Regressors

Context regression trackers can be generally categorized as either siamese or DCF methods. In the siamese methods, an exemplar image patch $\mathbf{z}$ and a search image patch $\mathbf{x}$ are defined. Here, $\mathbf{z}$ contains the target object and limited surrounding context, and $\mathbf{x}$ represents a target searching area,

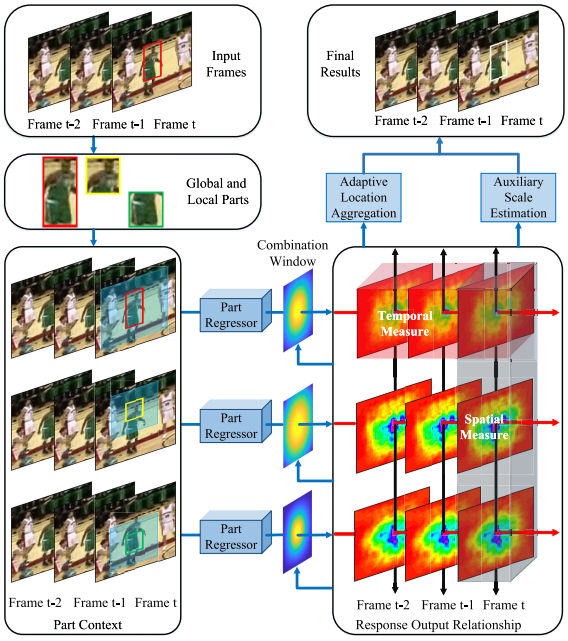

Fig. 2. The pipeline of our part-aware framework. To be collaboratively aware of the target state online, we first generate multiple parts containing one global and several local parts and apply context regressors to them. The response maps used to estimate the bounding box centers of parts denote the outputs of part regressors. Next, the tracking quality of each part regressor is scored by using the spatial-temporal measure to fully explore the relationship of response outputs, and the coarse target locations predicted by part regressors as well as these scores are aggregated to estimate the final target location. Then, these scores and response outputs are further utilized to construct combination window functions in part regressors for adaptively filtering redundant noise in the next frame. Finally, the target state is decided by jointing the target location and scale, which is estimated by additionally considering the spatial-temporal output information among multiple part regressors.

which is usually large. Both patches as inputs are fed to the weight-sharing CNN $\Psi(\cdot)$, of which feature maps are cross-correlated to generate the response output $\mathbf{r}$ as follows:

$$\mathbf{r}(\mathbf{z}, \mathbf{x}) = \Psi(\mathbf{z}) * \Psi(\mathbf{x}) + b \cdot \mathbb{I} \tag{1}$$

where $*$ denotes the cross-correlation operation, $b \cdot \mathbb{I}$ is the bias term. The convolutional kernel $\Psi(\mathbf{z})$ for template matching is the key to improving the robustness of siamese networks.

In another context regressor trackers, the tracking model $\mathbf{f}$ is optimized by hand-crafted or CNN features based DCF methods with a ridge regression formulation as follows:

$$\min_{\mathbf{f}} \|\mathbf{f} * \Psi(\mathbf{z}^{\star}) - \mathbf{y}\|_2^2 + \lambda \|\mathbf{f}\|_2^2 \tag{2}$$

where $\mathbf{y}$ denotes the ground-truth labels from the Gaussian distribution corresponding to the template image patch $\mathbf{z}^{\star}$ and λ is the regularization coefficient to prevent overfitting. It is to note that the template image patch $\mathbf{z}^{\star}$ in DCF methods contains more surrounding context and is much larger than the exemplar image patch $\mathbf{z}$ in siamese methods. In other words, DCF methods exploit more background information to discriminate the target compared to siamese methods. Finally, the response output is generated by $\mathbf{r} = \mathbf{f} * \Psi(\mathbf{x})$ after training the tracking model $\mathbf{f}$. The above ridge regression problem is solved by the closed-form solution in the Fourier domain [14] or the end-to-end CNN manner with the stochastic gradient descent [21] or the conjugate gradient approach [15].

To the end, both siamese and DCF trackers identify the location of the target by searching for the maximum value of $\mathbf{r}$. However, as mentioned earlier, most of them usually explore the holistic target representation with a single global regressor, which is prone to model drift in the challenging scenes of partial occlusion, deformation and drastic appearance variations due to ignoring the structural representation of the target appearance. To alleviate this problem, we decompose the target appearance into multiple parts, where the one is the global part of the target p, $p = 1$ and the others are the relatively arranged local parts of the target p, $p \in 2, \ldots, N$. Obviously, the global part pays attention to the changes of the entire target appearance in challenging scenarios such as fast move and scale variation, while the local parts focus on the important information of non-occluded target appearance when partial occlusion or deformation occurs. More specifically for local parts, they apply a vertically aligned spatial layout when the height of the target is greater than its width as shown in Figure 2, otherwise they are arranged in a horizontal alignment. The scales and locations of the local parts are strictly confined to the global part of the target in each frame.

Then, the context regressors are applied to gain the response outputs of global and local parts. Note that these regressors for predicting response outputs are treated as black boxes, where the offline training and online update phases are the same as that of the original context regressors. Besides, since the relative motions between these parts are few in most scenes, it is reasonable that coarse target predictions of all part regressors should be combined to refine the final target state. Similar to [62], [64], and [65], some parts with reliable information to locate the target should have the larger weights, while the unstable ones (*e.g.*, the occluded parts) are assigned the smaller weights. But different from [62], [64], and [65] using a relatively straightforward weight definition method, we make full use of the relationship of outputs among part regressors and propose the spatial-temporal measure to define the weight for each part, of which the details are presented in the following sections.

B. Spatial-Temporal Measure

As shown in Figure 2, multiple part regressors keep track of the target in parallel. In each frame of tracking, the measure in the spatial domain among different response outputs reveals the consistency degree among part regressors, which is defined as the spatial measure. In addition, the response output of each part regressor should be measured by its smoothness and continuity in the temporal domain. Therefore, given a response output, the tracking quality of the related part regressor can be represented by the spatial-temporal measure. After achieving the overall measure of each response output, the coarse target predictions of all part regressors are adaptively aggregated to refine the final tracking results by treating the spatial-temporal measure of each part as the weight. In the following, we introduce the formulation of spatial measure, temporal measure and adaptive location aggregation.

1) Spatial Measure: Most part regressors in the part-aware framework are able to keep track of the target robustly

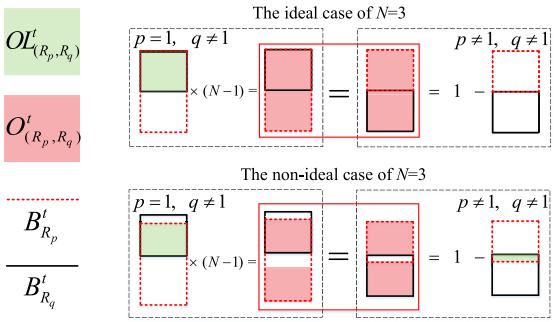

Fig. 3. An illustration of the output imbalance problem among the bounding boxes of global and local parts. The top row shows the ideal spatial arrangement case of one global part and two local parts while the bottom row denotes the non-ideal one where the irregular arrangement of parts often happens during tracking. The overlap areas (light green) among the bounding boxes of global and local parts on both sides in each case are not equal while the converted overlap areas (light red) in the middle of each case are equal.

in the general scenarios and the target location predicted by an ideal part regressor should be consistent with that of the other part regressors as much as possible, which requires that the bounding box outputs of these parts are relatively compact. Let $R_1, \ldots, R_p, \ldots, R_N$ represent Regressor $1, \ldots,$ Regressor$p, \ldots,$ Regressor$N$, which correspond to multiple parts of the target in our framework respectively. We define $B^t_{R_p}$ to denote the bounding box output by Regressor p in the t-th frame and treat all part regressors as black boxes. It is noted that only the location and size details about the part state are contained in the bounding box output $B^t_{R_p}$ without any surrounding information. Then, the overlap ratios among the bounding boxes from different part regressors are computed to measure the tracking quality of part regressors in the spatial domain. We first denote the overlap ratio $OL^t_{(R_p,R_q)}$ of Regressor p and Regressor q at frame t as follows:

$$OL^t_{(R_p,R_q)} = \frac{Area(B^t_{R_p} \bigcap B^t_{R_q})}{Area(B^t_{R_p} \bigcup B^t_{R_q})} \tag{3}$$

However, it is not ideal to directly compute the overlap ratios by taking the real overlap area between global and local parts with equal weights. For example, assuming the ideal case of $N = 3$ for clarity, the arrangement of parts is strictly restricted as shown in Figure 3. The overlap ratio denoted by the green area between global and local parts is 50% while the one between local parts is 0%, thus leading to the output imbalance problem among the bounding boxes of global and local parts when using the above equation to calculate overlap ratios of all parts. In order to avoid this problem, we convert their overlap ratios $OL^t_{(R_p,R_q)}$ into a unified calculation space by the following expression:

$$O^t_{(R_p,R_q)} = \begin{cases} (N-1)OL^t_{(R_p,R_q)} & if\ p \mid q = 1; \\ 1 - OL^t_{(R_p,R_q)} & otherwise; \end{cases} \tag{4}$$

In this way, as shown in the ideal case of $N = 3$ of Figure 3, the new overlap ratio $O^t_{(R_p,R_q)}$ represented by the red area between global and local parts is 100%, which is the same as the one between local parts. Obviously, this conversion is also suitable for the non-ideal cause of $N = 3$ in Figure 3, where the irregular arrangement of parts often occurs during tracking. Besides, we use a nonlinear Gaussian function in $O^t_{(R_p,R_q)}$ to avoid the large fluctuations among overlap ratios:

$$O'^t_{(R_p,R_q)} = exp(-(1 - O^t_{(R_p,R_q)})^2) \tag{5}$$

The mean overlap ratio between Regressor p and all other part regressors is computed by $M^t_{R_p} = \frac{1}{K}\sum_{q=1}^{K} O'^t_{(R_p,R_q)}$, which is used to describe the trajectory consistency of part regressors. Here, K represents the number of part regressors. Generally speaking, the mean overlap ratio in each frame should be stable. Thus, the stability of overlap ratios between R_p and other part regressors can be described by their fluctuation extent in a short period of time $\triangle t$ (*e.g.*, 5 frames), which is expressed by:

$$V^t_{R_p} = \sqrt{\frac{1}{K}\sum_{q=1}^{K}(O'^t_{(R_p,R_q)} - \frac{1}{\triangle t}\sum_{\tau=1}^{\triangle t} O'^{\tau+t-\triangle t}_{(R_p,R_q)})^2} \tag{6}$$

Here, we compute the weighted mean values of $M^t_{R_p}$ and $V^t_{R_p}$ over $\triangle t$ frames by introducing an increasing weight sequence $\mathbf{W} = \xi^0, \xi^1, \ldots, \xi^{\triangle t-1}, (\xi > 1)$ to the recent measures, which can avoid the output fluctuation of regressors:

$$M'^t_{R_p} = \frac{\sum_{\tau=1}^{\triangle t} W_\tau M^{\tau+t-\triangle t}_{R_p}}{\sum_{\tau=1}^{\triangle t} W_\tau} \tag{7}$$

$$V'^t_{R_p} = \frac{\sum_{\tau=1}^{\triangle t} W_\tau V^{\tau+t-\triangle t}_{R_p}}{\sum_{\tau=1}^{\triangle t} W_\tau} \tag{8}$$

Here, the $\tau + t - \triangle t$-th element in sequence $\mathbf{W}$ is denoted by W_τ. Finally, we define the spatial measure of Regressor p at the t-th frame as follows:

$$E^t_{sm}(R_p) = \frac{M'^t_{R_p}}{V'^t_{R_p} + \kappa} \tag{9}$$

Here, κ is a small constant that prevents the infinite spatial measure for a zero denominator. The larger the value of $E^t_{sm}(R_p)$ indicates the better consistency between Regressor p and other regressors and the higher the tracking quality.

2) Temporal Measure: Similarly, the response output smoothness degree of each part regressor in the temporal domain also can reveal the stability of its tracking performance to some extent. Therefore, we first compute the Euclidean distance measuring the shift between the current response output $\mathbf{r}^t_{R_p}$ and the previous response output $\mathbf{r}^{t-1}_{R_p}$. Then, we apply the nonlinear Gaussian function to alleviate the output imbalance problem among the response maps of global and local parts because of their different sizes, and define the temporal measure of Regressor p as follows:

$$E^t_{tm}(R_p) = \frac{1}{\sigma_{R_p}} exp(-\theta \|\mathbf{r}^t_{R_p} - \mathbf{r}^{t-1}_{R_p}\|^2) \tag{10}$$

Here, θ is a small constraint parameter. The larger the value of $E^t_{tm}(R_p)$ represents the better response smoothness. To the end, the spatial-temporal measure $E^t(R_p)$ of Regressor p at

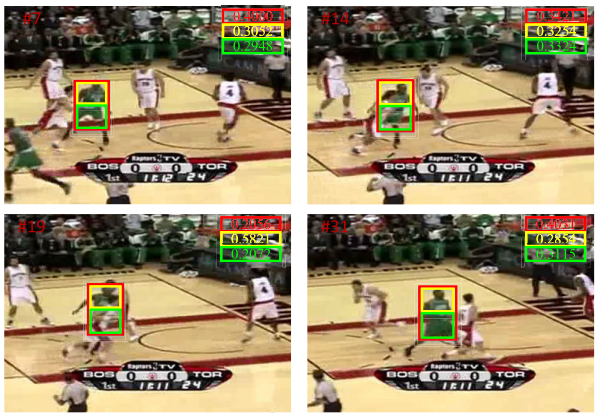

Fig. 4. Adaptive location aggregation process of one of our PAF trackers in *Basketball* sequence. The upper right corner of each video frame shows the normalized spatial-temporal measure scores of different part regressors. Some parts with reliable information to locate the target have the larger scores, while the unstable ones (*e.g.*, the occluded parts) are assigned the smaller scores.

the frame t is constructed by a linear fusion of its spatial measure $E^t_{sm}(R_p)$ and temporal measure $E^t_{tm}(R_p)$:

$$E^t(R_p) = \mu E^t_{sm}(R_p) + (1-\mu)E^t_{tm}(R_p) \tag{11}$$

Here, μ denotes the trade-off parameter between the spatial and temporal measures.

3) Adaptive Location Aggregation: By treating the spatial-temporal measures of each part regressor as the weights, the proposed framework can refine the final target location $C(B^{'t})$, which is also the center of the bounding box $B^{'t}$ of the target at the frame t. Based on the part location $C(B^t_{R_p})$ predicted by Regressor p in the t-th frame and the displacement vector $\nabla^{t-1}_{R_p}$, which is the length and direction of the vector between part p and the target center in the previous frame, we adaptively aggregate the coarse predictions of the target location $C(B^t_{R_p}) + \nabla^{t-1}_{R_p}$, $p \in 1, \ldots, N$ to estimate the final target location $C(B^{'t})$ by the following expression:

$$C(B^{'t}) = \frac{\sum_{p=1}^{N} E^t(R_p)[C(B^t_{R_p}) + \nabla^{t-1}_{R_p}]}{\sum_{p=1}^{N} E^t(R_p)} \tag{12}$$

Figure 4 shows the adaptive location aggregation process in the case of $N = 3$, where the parts with reliable information are assigned the larger weights while the unstable ones (*e.g.*, the occluded parts) have the smaller weights in each frame.

C. Combination Window Function

Owing to the adaptive aggregation mechanism for refining the target location, the window function used to locate the target should be carefully considered to alleviate the corruption of part regressors. The cosine window is widely used in context regressors including siamese and DCF methods to fight against noise interference and boundary effects. Nevertheless, its shape is fixed for various targets and does not change with significant noise variations during tracking, thus easily leading to the corruption of regressors. To this end, as shown in Figure 5, we propose to adaptively filter dynamic noise for each part regressor by the combination window function

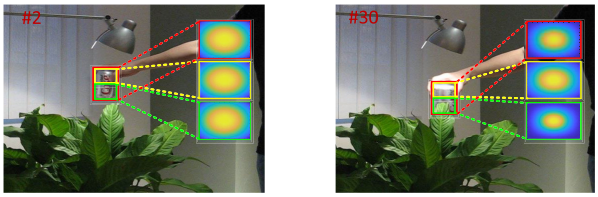

Fig. 5. An illustration of the combination window in each part regressor, which consists of the constant window (left) and the variation window (right). The constant windows are enough to efficiently suppress the noise undergoing general challenging scenarios (*e.g.*, illumination variations) while the variation windows are activated to adaptively change their shape to filter the redundant noise in more challenging scenarios (*e.g.*, drastic appearance variations).

including the constant and variable windows, of which both are built by the Kaiser window function suggested in [67]:

$$KW(j) = \begin{cases} \dfrac{I_0(\eta\sqrt{1-(\frac{2j}{J-1})^2})}{I_0(\eta)}, & for\ j = 1\ to\ J \\ 0, & elsewhere \end{cases} \tag{13}$$

where $I_0(\eta)$ denotes the modified Bessel function of the first kind of order zero $I_0(\eta) = 1 + \sum_{l=1}^{\infty}[\frac{1}{l!}(\frac{\eta}{2})^l]^2$ and J is the window scale. The trade-off between the main-lobe width and the side-lobe amplitude can be controlled by the parameter η. In other words, the parameter η can adjust the shape of the suppression noise region in the Kaiser window to filter the redundant noise. For more detail, please refer to [67].

On the other hand, many tracking algorithms [54], [62], [63] adopt the online updates of samples during tracking by using the peak-to-sidelobe ratio (PSR) to quantify the degree of noise interference. In contrast to them, we explore the PSR based on multiple parts to control the parameter η of the Kaiser window. PSR is given as $PSR = [max(\mathbf{r}) - \mu_\phi(\mathbf{r})]/\sigma_\phi(\mathbf{r})$, where μ_ϕ is the mean and σ_ϕ denotes the standard deviation of the response output $\mathbf{r}$. Then, we compute the average PSR of different parts $PSR^t_{mean} = \frac{1}{N}\sum_{p=1}^{N} PSR^t(R_p)$, where $PSR^t(R_p)$ denotes the PSR values of the response outputs of Regressors p. Besides, in order to make full use of the spatial-temporal relationship of multiple part regressors, we adopt a combined noise interference score $I^t = PSR^t_{mean} \cdot E^t_{mean}$ by considering the average PSR PSR^t_{mean} and the average spatial-temporal measure of part regressors $E^t_{mean} = \frac{1}{N}\sum_{p=1}^{N} E^t(R_p)$ together to evaluate the degree of noise interference. The lower the score means the more serious the noise interference.

When the current noise score I^t is lower than the past average noise score $I^{1:t}_{mean} = \frac{1}{t}\sum_{\tilde{t}=1}^{t} I^{\tilde{t}}$, the variable window in the proposed combination window is activated for each part regressor and its control parameter $\eta(R_p)$ is described by:

$$\eta(R_p) = PSR^1(R_p)/PSR^t(R_p), \quad if\ I^t < \alpha \cdot I^{1:t}_{mean} \tag{14}$$

where α denotes the noise threshold and $PSR^1(R_p)$ is the PSR value of Regressorr p in the first frame. Besides, when the noise score I^t is higher than the average noise score $I^{1:t}_{mean}$, the constant window with the invariant parameter $\eta(R_p) = \zeta$ in the combination window is switched for all part regressors. Therefore, the designed combination window function penalizes the response outputs with low scores severely to adaptively filter the redundant noise.

D. Auxiliary Scale Estimation

Another problem for context regression trackers is scale estimation. Despite showing remarkable scale estimation results based on the global target appearance, most of them ignore the scale information based on local parts of the target. In addition to the popular multi-scale search [41] or bounding box regression [11], [16], spatial-temporal output information among multiple part regressors is also leveraged to assist in estimating the target scale. The proposed auxiliary scale estimation is reasonable because the scale changes between adjacent frames are small and smooth in most scenes.

In the suggested method, similar to the work in [65], two parts have high weights and move further away from each other, which indicates an increase in the target size. Otherwise, two reliable parts moving closer to each other denote the decreasing of the target scale. Thus, we compute the Euclidean distance measuring the length among different parts $\|C(B^t_{R_p}) - C(B^t_{R_q})\|$, where $C(B^t_{R_p})$ is the predicted part location at the frame t. Besides, estimating the target scale in each frame ignoring the previous target size information may lead to inaccurate predictions, we also consider the previous distance $\|C(B^{'t-1}_{R_p}) - C(B^{'t-1}_{R_q})\|$, where $C(B^{'t-1}_{R_p})$ is the updated part location by the feedback of the final target state in the previous frame. Then, according to the spatial-temporal output information, we set a scale factor S^t_{sf} only if the target size shows a constant decrease or increase trend as follows:

$$S^t_{sf} = \frac{2}{N(K-1)} \sum_{p=1}^{N} \sum_{q=1}^{K} \Big[\frac{\|C(B^t_{R_p}) - C(B^t_{R_q})\|^2}{\|C(B^{'t-1}_{R_p}) - C(B^{'t-1}_{R_q})\|^2} \cdot exp(-\frac{1}{\epsilon^2}\|SV^t_{R_p} - SV^t_{R_q}\|^2) \Big], \quad (p \neq q) \tag{15}$$

Here, $SV^t_{R_p}$ denotes the shift vector decided by the response outputs of Regressor p and ϵ is a constant about the target size. Finally, the target size is decided by combining the native scale estimation results $S(B^t_{R_{p=1}})$ of context regression trackers (*e.g.*, siamese or DCF) based on the global part of the target: $S(B^{'t}) = (1-\nu) \cdot S(B^t_{R_{p=1}}) + \nu \cdot S^t_{sf} \cdot S(B^{'t-1})$, where ν is a trade-off constant between the native and auxiliary scale estimation. To the end, we combine $C(B^{'t})$ and $S(B^{'t})$ to decide the target state $B^{'t}$, which is as feedback to update the spatial arrangement of part regressors in the next frames. The whole tracking framework is summarized in Algorithm 1.

Algorithm 1 Part-Aware Framework

Input: Image IM^t, Target state $B^{'t-1}$, Part regressor R_p, Combination window $\mathbb{C}^t_{R_p}$;
Output: Target state $B^{'t}$, Combination window $\mathbb{C}^{t+1}_{R_p}$;
for $IM^t = 1 : t$ **do**
  if $t > 1$ **then**
    // Global and local part regressors
    Build global and local parts based on $B^{'t-1}$;
    Crop the part context by the setting of $\{R_p\}^N_{p=1}$;
    Get the response outputs $\{\mathbf{r}^t_{R_p}\}^N_{p=1}$ of all parts using $\{R_p\}^N_{p=1}$ and $\{\mathbb{C}^t_{R_p}\}^N_{p=1}$;
    // Spatial-temporal measure
    Compute the spatial-temporal measure score $\{E^t(R_p)\}^N_{p=1}$ by Eq. 3 - Eq. 11;
    Estimate the target location $C(B^{'t})$ using Eq. 12;
    // Auxiliary scale estimate
    Infer the target size $S(B^{'t})$ by merging $S(B^t_{R_{p=1}})$ and $S^t_{sf} \cdot S(B^{'t-1})$ with Eq. 15;
    // Combination window function
    Evaluate the noise interference score I^t using PSR^t_{mean} and E^t_{mean} of $\{R_p\}^N_{p=1}$;
    if $I^t < \alpha \cdot I^{1:t}_{mean}$ **then**
      Construct the variable window of $\{\mathbb{C}^{t+1}_{R_p}\}^N_{p=1}$ with Eq. 13 and Eq. 14;
    end
    if $I^t \geqslant \alpha \cdot I^{1:t}_{mean}$ **then**
      Use the constant window of $\{\mathbb{C}^{t+1}_{R_p}\}^N_{p=1}$ with $\eta(R_p) = \zeta$ and Eq. 13;
    end
  end
  if $t = 1$ **then** initialize $PSR^t(R_p)$, $\mathbf{r}^t_{R_p}$ and the spatial arrangement parameters of $\{R_p\}^N_{p=1}$ **end**;
end

IV. Experiments

To comprehensively verify the effectiveness of the proposed scheme, we extend our part-aware framework (PAF) into many popular siamese and DCF trackers using the context regression scheme. More specifically, according to different scale estimation strategies, these trackers can also be summarized into two categories: multi-scale search and bounding box regression. Moreover, numerous experiments are performed on the large-scale benchmarks to prove the generalization ability of PAF, involving OTB [6], UAV [8], TC128 [7], UAVDT [18], LaSOT [9], TrackingNet [20], GOT-10k [19] and VOT [29], [30]. We follow their protocols and use two evaluation metrics in one-pass evaluation (OPE). One is the average distance precision (DP) denoting the rate of frames whose the center location is within 20 pixels of ground-truth locations and the other is the area-under-curve (AUC) indicating the overlap percentage between ground-truth bounding boxes and tracking outputs. In addition, different from other benchmarks, VOT resets the trackers to the ground-truth locations when tracking fails. Therefore, the expected average overlap (EAO) is applied to evaluate the tracking performance, representing the inner product between the predicted average overlap and the sequence-length distribution and measuring the expected no-reset overlap on a video.

A. Implementation Details

All trackers are executed on an Intel Xeon E5-2640 v4 2.4GHz CPU with 128 GB RAM and a single GeForce GTX TITAN XP GPU. The trackers with multi-scale search adopt the MatConvNet toolbox in Matlab while the trackers with bounding box regression use PyTorch in Python. Based on the

TABLE I

MULTI-SCALE SEARCH BASELINES INTEGRATED TO OUR PAF

Siamese Trackers	Where/When	DCF Trackers	Where/When
SiamFC [10]	ECCVW/2016	Staple [50]	CVPR/2016
CFNet [32]	CVPR/2017	BACF [51]	ICCV/2017
Triplet [33]	ECCV/2018	STRCFH [68]	CVPR/2018
TADT [38]	CVPR/2019	ARCFH [69]	ICCV/2019
UDT [41]	IJCV/2021	AutoTrackH [52]	CVPR/2020

above environment, we measured the time spent on computing the output of each frame and give the mean FPS over the OTB [6] benchmark. For a fair comparison, all compared trackers are run with the same hyper-parameters or tracking results offered by their authors. The number of parts N and the parameter ξ in the weigh sequence $\mathbf{W}$ are set to 3 and 1.1. The small constants κ and θ in Eq. 9 and Eq. 10 are 0.008 and 0.0001, respectively. The trade-off parameter μ in Eq. 11 is set to 0.1. Additional parameters in the proposed trackers and experimental results are left in the supplementary material.

B. Multi-Scale Search Baseline Comparison

1) Extend Our Framework to Multiple Baselines: In the early context regression based algorithms, most siamese and DCF trackers adopt multi-scale search to estimate target sizes, which are selected as our baselines to express the generalization of our PAF. In siamese trackers, many extensions [32], [33], [38], [41] are based on SiamFC [10], which is chosen as the baseline of our PAF. These extensions are also chosen as our baselines, including CFNet [32] integrating correlation filters, Triplet [33] applying triplet loss, TADT [38] learning target-aware features and UDT [41] training unsupervised features. In DCF trackers, many improvements are based on DCF [14] and some of them with multi-scale search and HOG features are chosen as the baselines for a fair comparison, like Staple [50] complementing color information, BACF [51] considering background information, STRCFH [68] adding spatial-temporal constraints, ARCFH [69] suppressing response aberrances and AutoTrackH [52] using automatic spatial-temporal constraints. Table I summarizes the above siamese and DCF trackers in detail. We apply the proposed PAF to these multi-scale search baselines and refer to them as *_PAF, where * represents the baseline name.

2) Evaluate PAF Trackers on Multiple Benchmarks: To validate the generalization of our PAF, the evaluations with their baselines are implemented on five classic benchmarks.

a) OTB: The OTB benchmark [6] contains OTB2013 with 51 videos, OTB2015 with 100 videos and OTB50 with 50 more challenging videos. DP and AUC of baselines and PAF counterparts in OPE on OTB are shown in Table II. Besides, the performance gains and the tracking speed changes (*e.g.*, FPS tested with CPU or GPU) by extending our PAF to these baselines are also provided. We can see that the proposed PAF helps all baselines achieve performance improvements with a large margin on DP and AUC. TADT_PAF ranks *1st* among all trackers, and SiamFC_PAF obtains the biggest improvement 7.2% DP on OTB50 in siamese trackers while the biggest improvement in DCF trackers comes from 6.9% DP at Staple_PAF in OTB2013. Note that even using multiple

TABLE II

DP(%), AUC(%) AND FPS ON OTB. WE DENOTE THE BEST RESULTS IN **BLUE** ON EACH PART. ± DENOTES THE PERFORMANCE GAIN (**RED** REPRESENTS INCREASE AND **GREEN** MEANS DECREASE)

	Trackers	OTB2013 [6] DP	OTB2013 [6] AUC	OTB50 [6] DP	OTB50 [6] AUC	OTB2015 [6] DP	OTB2015 [6] AUC	Mean FPS
Siamese network	SiamFC [10]	80.1	60.6	69.1	51.5	76.9	58.1	**82.4@GPU**
	SiamFC_PAF	**85.1**	**64.0**	**76.3**	**55.8**	**80.8**	**60.7**	44.7@GPU
	±	**+5.0**	**+3.4**	**+7.2**	**+4.3**	**+3.9**	**+2.6**	**-37.7@GPU**
	CFNet [32]	80.7	61.1	70.2	53.0	74.8	56.8	**53.4@GPU**
	CFNet_PAF	**85.5**	**64.5**	**75.7**	**56.6**	**80.7**	**60.8**	29.9@GPU
	±	**+4.8**	**+3.4**	**+5.5**	**+3.6**	**+5.9**	**+4.0**	**-23.5@GPU**
	Triplet [33]	82.1	61.7	71.3	52.7	77.7	58.7	**75.1@GPU**
	Triplet_PAF	**85.5**	**64.2**	**77.3**	**56.1**	**81.9**	**61.2**	43.5@GPU
	±	**+3.4**	**+2.5**	**+6.0**	**+3.4**	**+4.2**	**+2.5**	**-31.6@GPU**
	TADT [38]	89.7	68.0	83.3	61.8	86.4	65.8	**77.0@GPU**
	TADT_PAF	**90.5**	**69.2**	**84.0**	**62.5**	**87.2**	**66.7**	27.3@GPU
	±	**+0.8**	**+1.2**	**+0.7**	**+0.7**	**+0.8**	**+0.9**	**-49.7@GPU**
	UDT [41]	82.0	62.2	67.7	51.8	76.0	58.7	**119.6@GPU**
	UDT_PAF	**84.0**	**64.0**	**70.3**	**53.2**	**78.5**	**60.4**	41.9@GPU
	±	**+2.0**	**+1.8**	**+2.6**	**+1.4**	**+2.5**	**+1.7**	**-77.7@GPU**
Discriminant correlation filter	Staple [50]	79.3	60.0	68.1	50.9	78.4	58.2	**65.0@CPU**
	Staple_PAF	**86.2**	**64.3**	**72.6**	**53.6**	**78.7**	**58.5**	32.5@CPU
	±	**+6.9**	**+4.3**	**+4.5**	**+2.7**	**+0.3**	**+0.3**	**-32.5@CPU**
	BACF [51]	83.0	64.4	74.3	56.6	81.2	61.9	**30.1@CPU**
	BACF_PAF	**86.4**	**66.0**	**77.0**	**57.6**	**83.4**	**63.0**	14.9@CPU
	±	**+3.4**	**+2.4**	**+2.7**	**+1.0**	**+2.2**	**+1.1**	**-15.2@CPU**
	STRCFH [68]	82.6	64.4	77.7	58.6	84.1	64.5	**31.8@CPU**
	STRCFH_PAF	**86.8**	**67.0**	**78.0**	**58.7**	**85.1**	**65.3**	18.3@CPU
	±	**+4.2**	**+2.6**	**+0.3**	**+0.1**	**+1.0**	**+0.8**	**-13.5@CPU**
	ARCFH [69]	85.2	63.6	**75.7**	**54.8**	80.7	59.8	**27.6@CPU**
	ARCFH_PAF	**86.5**	**65.1**	73.9	53.7	**82.2**	**61.6**	14.1@CPU
	±	**+1.3**	**+1.5**	**-1.8**	**-1.1**	**+1.5**	**+1.8**	**-13.5@CPU**
	AutoTrackH [52]	84.1	61.9	74.2	53.0	77.2	58.2	**50.8@CPU**
	AutoTrackH_PAF	**87.3**	**65.1**	**75.7**	**55.4**	**83.1**	**62.4**	23.0@CPU
	±	**+3.2**	**+3.2**	**+1.5**	**+2.4**	**+5.9**	**+4.2**	**-27.8@CPU**

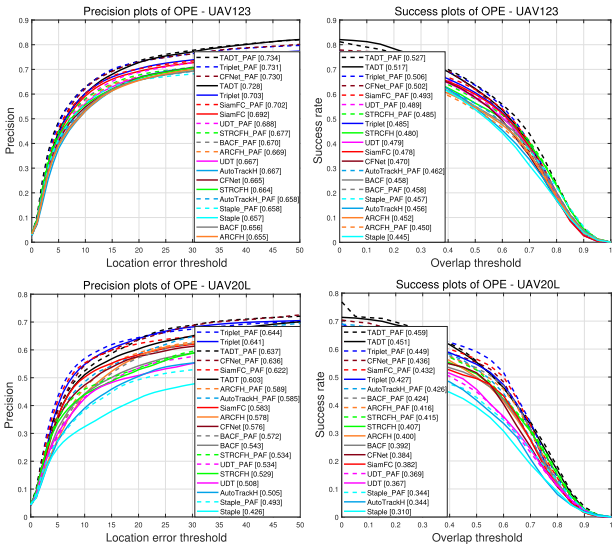

Fig. 6. DP and AUC comparison of multi-scale search baselines on UAV.

parts of the target, all PAF trackers with the siamese network are at least with the real-time speed of 27 FPS on GPU.

b) UAV: There are UAV123 with 123 low altitude aerial videos and UAV20L with 20 long-term videos in the UAV benchmark [8]. Figure 6 shows the results of all trackers over UAV123 and UAV20L. Despite UAV is more challenging than OTB, the performance improvement with PAF has not been affected and most PAF trackers consistently outperform their corresponding baselines. Specifically, TADT_PAF achieves the top performance on AUC and DP over UAV123. As for UAV20L, Triplet_PAF ranks *1st* and achieves the performance improvement of 0.3% compared to Triplet on DP.

c) TC128: The TC128 benchmark [7] includes 128 challenging color videos. The tracking results of baselines and PAF trackers over TC128 are shown in Figure 7. As can be seen, most baselines achieve performance improvement

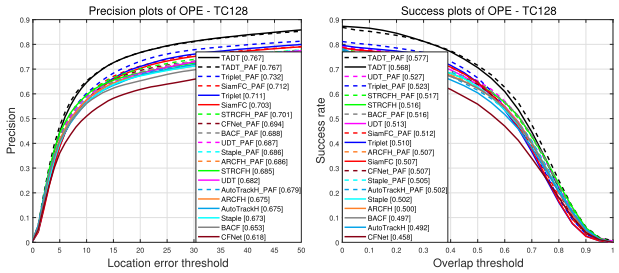

Fig. 7. DP and AUC comparison of multi-scale search baselines on TC128.

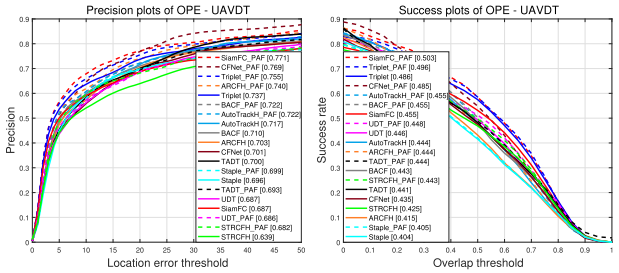

Fig. 8. DP and AUC comparison of multi-scale search baselines on UAVDT.

by adopting our PAF, which is the same as the previous assessment. Especially for CFNet_PAF, the 7.6% and 4.9% significant improvement in terms of DP and AUC are obtained, even though TC128 has more challenging videos than OTB. Moreover, TADT_PAF with 57.7% AUC performs favorably against the other trackers.

d) UAVDT: The UAVDT benchmark [18] consists of 100 video sequences captured from the UAV platform. We show DP and AUC of baselines and PAF counterparts with OPE on UAVDT in Figure 8. As can be seen, our PAF enables many siamese and DCF trackers to achieve performance improvements on UAVDT. In particular, SiamFC_PAF in siamese trackers, of which DP and AUC are 8.4% and 4.8% better than that of SiamFC, performs favorably against the others due to handling drastic appearance variations well by our proposed PAF.

e) LaSOT: The baseline comparison experiments are also implemented on the very recent and most challenging LaSOT [9] benchmark, consisting of 1,120 training videos and 280 testing videos. As shown in Figure 9, DP and AUC of all trackers on the testing videos of LaSOT achieve the lowest performance among the above benchmarks. Nonetheless, our PAF can also work to improve the tracking performance of baselines. Most siamese trackers with our PAF perform favorably against the DCF trackers whether or not using the proposed PAF. Furthermore, TADT_PAF achieves the highest 36.8%/35.9% in AUC and DP among all trackers.

C. Bounding Box Regression Baseline Comparison

1) Extend Our Framework to Multiple Baselines: Until very recently, most context regression trackers are able to accurately estimate the target scale by bounding box regression, such as the popular SiamRPN++ [11] integrating the region proposal network into siamese trackers and the advanced DiMP [16] introducing IoUNet [15] to the deep network based DCF.

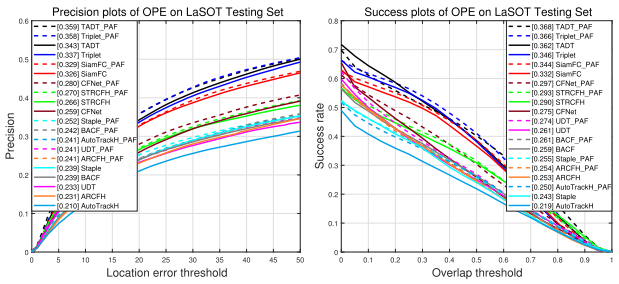

Fig. 9. DP and AUC comparison of multi-scale search baselines on LaSOT.

TABLE III
BOUNDING BOX REGRESSION BASELINES INTEGRATED TO OUR PAF

Siamese Trackers	Where/When	DCF Trackers	Where/When
SiamRPN++ [11]	CVPR/2019	DiMP [16]	ICCV/2019
SiamCAR [12]	CVPR/2020	TrDiMP [61]	CVPR/2021

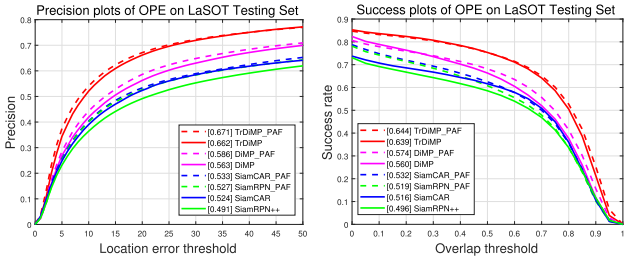

Fig. 10. DP/AUC comparison of boundingbox regression baselines on LaSOT.

TABLE IV
COMPARISON OF BOUNDING BOX REGRESSION BASELINES ON GOT-10K WITH AO(%) AND $SR_{0.5}$(%) AND TRACKINGNET WITH DP (%) AND AUC (%). FPS TESTED WITH GPU IS ALSO REPORTED. THE BEST RESULTS ARE HIGHLIGHTED IN EACH PART

	Trackers	GOT-10k [19] AO	GOT-10k [19] $SR_{0.5}$	TrackingNet [20] DP	TrackingNet [20] AUC	Mean FPS
Siamese	SiamRPN++ [11]	51.7	61.6	69.4	73.3	**35.4@GPU**
	SiamRPN_PAF	**53.6**	**63.9**	**69.4**	**74.4**	13.9@GPU
	SiamCAR [12]	58.1	68.3	68.4	74.0	**36.8@GPU**
	SiamCAR_PAF	**58.6**	**69.6**	**69.0**	**74.8**	15.5@GPU
DCF	DiMP [16]	61.1	71.7	68.7	74.0	**36.1@GPU**
	DiMP_PAF	**62.0**	**72.4**	**69.1**	**74.6**	15.2@GPU
	TrDiMP [61]	68.8	80.5	73.1	**78.4**	**22.3@GPU**
	TrDiMP_PAF	**69.6**	**80.7**	**74.3**	78.1	5.0@GPU

Therefore, we select them as our baselines to further prove the generalization of our PAF. Besides, their recent extensions are also chosen as our baselines, involving SiamCAR [12] designing an anchor-free siamese network with a classification and regression scheme, and TrDiMP [61] extending the transformer architecture to DiMP with the rich temporal information. These bounding box regression baselines are summarized in detail on Table III. We apply our PAF to these baselines and refer them as SiamRPN_PAF, DiMP_PAF, SiamCAR_PAF and TrDiMP_PAF, of which the average tracking speeds tested with GPU are 13.9, 15.2, 15.5 and 5.0 FPS in Table IV, respectively. Note that the combination window functions in our PAF are not applied in DiMP and TrDiMP since they rarely use the window function.

2) Evaluate PAF Trackers on Multiple Benchmarks: We implement PAF trackers evaluated with their bounding box regression baselines on four recent large-scale benchmarks.

a) LaSOT: We evaluate PAF trackers on the LaSOT testing set with 280 videos. Figure 10 shows DP and AUC of baselines and PAF counterparts in OPE. Overall, all baselines

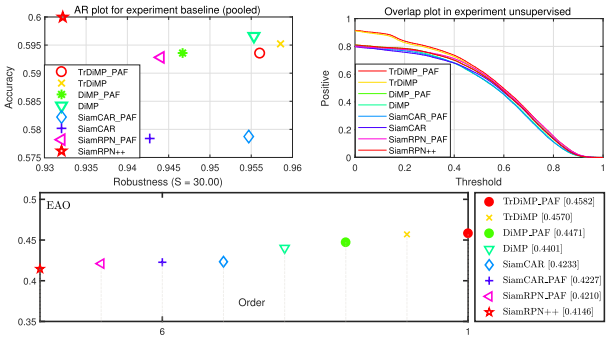

Fig. 11. Comparison of bounding box regression baseline trackers on VOT2018 with Accuracy, Robustness, unsupervised AO and EAO.

achieve the improvements on DP and AUC by our PAF. Specifically, SiamRPN_PAF obtains the biggest performance improvement 3.6% DP and 2.3% AUC in siamese trackers while the highest performance gain in DCF trackers comes from 2.3% DP and 1.4% AUC at DiMP_PAF. Besides, TrDiMP_PAF performs favorably against the others with 67.1% DP and 64.4% AUC.

b) VOT2018: Evaluations of baselines and PAF trackers are also conducted on VOT2018 [29] containing 60 challenging video sequences. Figure 11 shows Accuracy, Robustness, unsupervised Average Overlap (AO) and EAO scores of baselines and their PAF counterparts on VOT2018. As can be seen in EAO, most PAF trackers obtain superior performance than their respective baselines even under the reset mechanism. As for siamese trackers, even though SiamCAR_PAF is 0.05% lower than its baseline SiamCAR with EAO, the Robustness of SiamCAR_PAF is better than that of SiamCAR. As for DCF trackers, our PAF enables TrDiMP_PAF to achieve the top performance among all trackers on the unsupervised AO and obtain 0.12% improvement against TrDiMP [61] on EAO.

c) GOT-10k: There are more than 10000 challenging videos in GOT-10k [19], involving the testing set with 180 video sequences. Following its test protocol, we evaluate all trackers on testing videos with average overlap (AO) and success rates at overlap thresholds 0.5 ($SR_{0.5}$). As shown in Table IV, our PAF trackers still outperform their respective baselines. Specifically, TrDiMP_PAF ranks *1st* among all trackers and achieves performance gains of 0.8% on AO against TrDiMP. Especially, SiamRPN_PAF obtains 1.9% and 2.3% significant improvement on AO and $SR_{0.5}$ against SiamRPN++.

d) TrackingNet: TrackingNet [20] is a recently released large-scale dataset, which contains the testing set with 511 videos. The outputs of our PAF trackers are submitted to the online evaluation server, and the evaluations with DP and AUC are shown in Table IV. We can see that our PAF is also effective in improving the performance of these baselines. In particular, DP and AUC of SiamCAR_PAF in siamese trackers are 0.6% and 0.8% better than that of SiamCAR. Besides, although TrDiMP_PAF in DCF trackers is 0.3% lower than TrDiMP in AUC, DP of TrDiMP_PAF is 1.2% better than that of TrDiMP because of the ability to handle the local appearance variations.

TABLE V

COMPARISON WITH STATE-OF-THE-ART TRACKERS USING MULTI-SCALE SEARCH ON OTB2015, UAV123, AND LASOT IN DP (%) AND AUC (%). THE TOP THREE IN EACH PART ARE SHOWN IN **RED**, **BLUE** AND **GREEN**

	Trackers	Source	OTB2015[6] DP/AUC	UAV123[8] DP/AUC	LaSOT[9] DP/AUC
Multi-Scale Search	CCOT [58]	ECCV2016	89.8/67.1	73.2/51.7	-
	ECO [59]	ICCV2017	91.0/69.0	74.1/52.5	30.1/32.4
	DSLT [21]	ECCV2018	90.9/66.0	74.6/53.0	-
	AutoTrackH [52]	CVPR2020	77.2/58.2	66.7/45.6	21.0/21.9
	SiamFC [10]	ECCVW2016	76.9/58.1	69.2/47.8	33.6/33.2
	StructSiam [23]	ECCV2018	85.1/62.1	-	33.3/33.5
	GCT [34]	CVPR2019	85.4/64.8	73.2/50.8	-
	UDT [41]	IJCV2021	76.0/58.7	66.7/47.9	23.3/26.1
	STRCFH_PAF	Ours	85.1/65.3	67.7/48.5	27.0/29.3
	TADT_PAF	Ours	87.2/66.7	73.4/52.7	35.9/36.8

TABLE VI

COMPARISON WITH RECENT STATE-OF-THE-ART TRACKERS USING BOUNDING BOX REGRESSION ON OTB2015, UAV123, LASOT AND VOT2018 IN DP (%), AUC (%) AND EAO. THE TOP THREE IN EACH PART ARE SHOWN IN **RED**, **BLUE** AND **GREEN**

	Trackers	Source	OTB2015[6] DP/AUC	UAV123[8] DP/AUC	LaSOT[9] DP/AUC	VOT2018[29] EAO
	DiMP[16]	ICCV2019	89.9/68.7	85.8/64.8	56.7/56.9	0.440
Bounding box regression	PrDiMP[60]	CVPR2020	89.7/69.5	87.8/66.9	60.8/59.8	0.442
	TrDiMP[61]	CVPR2021	92.5/70.8	87.6/67.0	66.2/63.9	0.457
	SuperDiMP[57]	ICCV2021	90.5/70.1	88.5/67.1	65.3/63.1	-
	KeepTrack[17]	ICCV2021	92.2/70.9	90.0/68.3	70.2/67.1	-
	SiamRPN++[111]	CVPR2019	91.5/69.6	80.3/61.0	49.1/49.6	0.415
	AFOD[46]	ECCVW2020	-	-	-	0.491
	SiamBAN[44]	CVPR2020	91.0/69.6	83.3/63.1	52.1/51.4	0.452
	PACNet[71]	AAAI2021	87.6/67.0	82.7/62.0	-/55.3	-
	STMTrack[47]	CVPR2021	-/71.9	-/64.7	63.3/60.6	0.447
	SiamGAT[35]	CVPR2021	91.7/71.0	84.3/64.6	53.0/53.9	-
	SAOT[25]	ICCV2021	92.6/71.4	-	62.9/61.6	0.501
	AutoMatch[45]	ICCV2021	92.6/71.4	-	59.9/58.3	-
	SparseTT[70]	IJCAI2022	-/70.4	-/70.4	70.1/66.0	-
	SiamCAR_PAF	Ours	91.9/70.5	84.5/63.4	53.3/53.2	0.423
	TrDiMP_PAF	Ours	93.4/71.3	88.9/67.6	67.1/64.4	0.458

D. Comparison With State-of-the-Art Trackers

We compare our advanced PAF trackers (STRCFH_PAF, TADT_PAF, SiamCAR_PAF and TrDiMP_PAF) with the popular state-of-the-art trackers, such as multi-scale search trackers (UDT [41], AutoTrackH [52], GCT [34], ECO [59], etc.) and bounding box regression trackers (SparseTT [70], SAOT [25], KeepTrack [17], SuperDiMP [57], AFOD [46], etc.). The detailed results on OTB2015 [6], UAV123 [8], LaSOT [9] and VOT2018 [29] are reported in Table V and Table VI.

1) Comparison of Multi-Scale Search Trackers: Table V shows that the proposed TADT_PAF and STRCFH_PAF achieve competitive performance on AUC and DP compared to the state-of-the-art trackers using the multi-scale search, like UDT [41], AutoTrackH [52], ECO [59] and GCT [34]. In the recent LaSOT, STRCFH_PAF with 27.0% DP and 29.3% AUC performs favorably against UDT [41] and AutoTrackH [52]. Besides, TADT_PAF achieves performance gains of 2.6% and 3.3% on DP and AUC over LaSOT against StructSiam [23], which exploits the part-based approach during offline training.

2) Comparison of Bounding Box Regression Trackers: As shown in Table VI, TrDiMP_PAF outperforms most state-of-the-art trackers using bounding box regression with DP/AUC and EAO of 93.4%/71.3%, 88.9%/67.6%, 67.1%/64.4% and 0.458 on OTB2015, UAV123, LaSOT and VOT2018, respectively. Compared with the very recent nine methods [17], [25], [35], [45], [47], [57], [61], [70], [71], TrDiMP_PAF achieves the top three places in terms of AUC, DP and EAO. Besides, SiamCAR_PAF outperforms SiamBAN and PACNet using siamese networks with gains of 0.9%/0.9% and 4.3%/3.5%

TABLE VII

ANALYSIS OF EACH COMPONENT IN OUR PAF TRACKERS WITH DP AND AUC ON OTB2015, UAVDT, UAV123 AND LASOT. WE CAN SEE THAT PAF TRACKERS WITH ALL COMPONENTS ACHIEVE SUPERIOR PERFORMANCE. THE BEST RESULTS ARE HIGHLIGHTED IN EACH PART

Baseline with the global part			✓	✓	✓	✓
Spatial-temporal measure with multi-parts				✓	✓	✓
Auxiliary scale estimation					✓	✓
Combination window function						✓
Multi-scale search	OTB2015 [6]	SiamFC [10]	76.9/58.1	80.1/60.1	80.1/60.4	**80.8/60.7**
	DP/AUC	AutoTrackH [52]	77.2/58.2	80.7/61.2	82.7/62.2	**83.1/62.4**
	UAVDT [18]	SiamFC [10]	68.1/44.7	72.7/48.2	74.8/49.9	**77.1/50.3**
	DP/AUC	AutoTrackH [52]	71.7/44.4	71.5/45.1	71.6/45.2	**72.2/45.5**
Bounding box regression	UAV123 [8]	SiamRPN++ [11]	81.8/61.3	83.4/62.5	83.8/62.6	**84.0/62.8**
	DP/AUC	DiMP [16]	85.8/64.8	86.5/65.9	**86.7/66.2**	-
	LaSOT [9]	SiamRPN++ [11]	49.1/49.6	52.0/51.3	51.7/51.6	**52.7/51.9**
	DP/AUC	DiMP [16]	56.3/56.0	58.1/57.2	**58.6/57.4**	-

in DP and AUC over OTB2015, respectively. Overall, these results prove that TrDiMP_PAF and SiamCAR_PAF achieve comparable performance with state-of-the-art trackers.

E. Ablation Study

In this section, we conduct the ablation study of our representative PAF trackers using bounding box regression (SiamRPN_PAF and DiMP_PAF) and multi-scale search (SiamFC_PAF and AutoTrackH_PAF) with DP and AUC on UAV123 [8], LaSOT [9], OTB2015 [6] and UAVDT [18].

1) Component Analysis of PAF: To demonstrate the contributions of each component in our PAF, we implement and evaluate three more types of variations. First, we implement SiamRPN++ [11], DiMP [16], SiamFC [10] and AutoTrackH [52] as baselines. Second, we construct four trackers that only integrate the spatial-temporal measure with multi-parts into baselines, where their native scale estimations are used. Third, four trackers are performed by just applying the spatial-temporal measure and auxiliary scale estimation. The evaluation results are shown in Table VII. According to Table VII, the DP and AUC scores of baselines on all benchmarks increase when the spatial-temporal measure with multi-parts is added. Specially, the performance gains of SiamFC and AutoTrackH are 3.5%/3% and 4.6%/3.5% in DP/ AUC over OTB2015 and UAVDT while the DP/AUC scores of DiMP and SiamRPN++ increase to 86.5%/65.9% and 52.0%/51.3% from 85.8%/64.8% and 49.1%/49.6% in UAV123 and LaSOT. Similarly, when auxiliary scale estimation is continued to be adopted, the performance of trackers on all benchmarks increases by 0.1%-2%. For example, AutoTrackH achieves the performance improvement of 2.0%/1.0% in DP/ AUC over OTB2015 while the performance gains of DiMP are 0.2%/0.3% in DP/ AUC over UAV123. Finally, the DP and AUC scores of implemented trackers (except DiMP rarely adopts the window function) continue to increase when the combination window is adopted.

2) Part Number Analysis of PAF: Since adopting the part-based strategy, we provide the analysis of the number of parts on our PAF. Two more types of trackers are implemented. One is the baselines constructed with one global part, like SiamRPN++ [11], DiMP [16], SiamFC [10] and AutoTrackH [52]. The other one is variants of these PAF trackers with one global part and three local parts. Table VIII shows DP and AUC of implemented trackers with the different

TABLE VIII

EVALUATING OUR PAF TRACKERS WITH THE DIFFERENT NUMBERS OF GLOBAL AND LOCAL PARTS ON OTB2015 AND LASOT. RESULTS ARE REPORTED AS DP(%)/AUC(%) AND FPS IS ALSO PRESENTED FOR THE EFFICIENCY ANALYSIS. PAF TRACKERS WITH ONE GLOBAL AND TWO LOCAL PARTS ACHIEVE COMPETITIVE PERFORMANCE IN BOTH ACCURACY AND EFFICIENCY. THE BEST RESULTS ARE HIGHLIGHTED IN EACH PART

One global part			✓	✓	✓
Two local parts				✓	
Three local parts					✓
Multi-scale search	OTB2015 [6]	SiamFC [10]	76.9/58.1	**80.8/60.7**	78.9/59.5
	DP/AUC	AutoTrackH [52]	77.2/58.2	83.1/62.4	**83.2/62.5**
	Speed	SiamFC [10]	**82.4**	44.7	35.6
	FPS	AutoTrackH [52]	**50.8**	23.0	15.7
Bounding box regression	LaSOT [9]	SiamRPN++ [11]	49.1/49.6	**52.7**/51.9	51.7/**52.3**
	DP/AUC	DiMP [16]	56.3/56.0	**58.6/57.4**	57.1/56.8
	Speed	SiamRPN++ [11]	**35.4**	13.9	8.6
	FPS	DiMP [16]	**36.1**	15.2	9.1

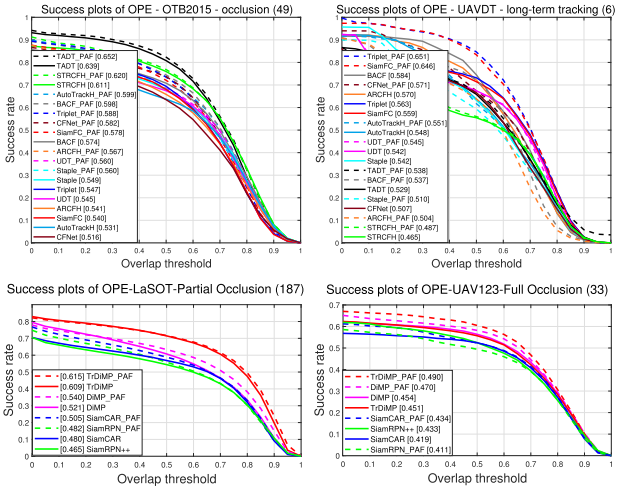

Fig. 12. Attribute-based AUC comparison of baseline trackers on occlusion, long-term tracking, partial occlusion and full occlusion.

numbers of parts over OTB2015 [6] and LaSOT [9]. FPS is also given to analyze the trade-off between efficiency and accuracy.

We can see that all trackers combining one global and two local parts perform favorably against their baselines with one global part, which proves increasing the number of local parts with a certain amount of computational cost in our PAF can improve the tracking performance. Besides, continuing to increase the number of local parts, such as one global and three local parts, may result in increased robustness but suffer in the tracking efficiency. For example in AutoTrackH and SiamRPN++, the AUC scores of one global part and three local parts are 0.1% and 0.4% better than that of one global and two local parts, but their tracking speed is reduced by 7.3 and 6.1 FPS. Therefore, considering a trade-off between accuracy and efficiency, we set the reasonable number of parts in our PAF to one global and two local parts, which also achieves superior performance and has the generalization.

F. Discussion

1) About Attribute: The AUC scores of four attributes are exhibited in Figure 12. In normal occlusion scenarios (occlusion, partial occlusion), all PAF trackers improve

TABLE IX
DETAIL COMPARISONS ON VOT2019 WITH THE STATE-OF-THE-ART IN TERMS OF EAO, ACCURACY AND ROBUSTNESS. THE BEST THREE RESULTS ARE SHOWN IN **RED**, **BLUE** AND **GREEN** FONTS

Tracker	SiamRPN++ [11]	SiamMask [43]	PACNet [71]	SiamRCR [40]	SiamRN [37]	Lighttrack [48]	DCFST [56]
EAO↑	0.285	0.287	0.303	0.336	0.341	0.357	0.361
Accuracy↑	0.599	0.594	0.573	0.602	0.593	0.552	0.589
Robustness↓	0.482	0.461	0.401	0.386	0.306	0.310	0.321

Tracker	SiamMargin [30]	SiamBANACM [39]	DiMP [16]	ATP [30]	SiamRPT [72]	SiamRPN_PAF Ours	DiMP_PAF Ours
EAO↑	0.362	0.362	0.379	**0.394**	**0.417**	0.296	**0.387**
Accuracy↑	0.578	**0.621**	0.594	**0.650**	**0.623**	0.606	0.615
Robustness↓	0.326	0.316	**0.278**	0.291	**0.186**	0.474	**0.259**

baselines by 0.6%-6.8% in AUC because PAF helps them fully exploit unoccluded target parts. In long-term tracking and full occlusion (disappearance, reappearance), most PAF trackers outperform their baselines, except for SiamRPN++, ARCFH, Staple and BACF. Like most context regressors [30], we believe our PAF can retrieve targets in these attributes by adding a redetection scheme, thereby steadily improving performance.

2) About Performance: Similarity based context regression tracking is an important research area. Recently, several powerful trackers [37], [39], [40], [43], [48], [56], [72] with context regressors (*e.g.*, DiMP or SiamRPN++) achieve excellent results on VOT2019 [30]. Table IX reports the detailed comparisons between these state-of-the-arts and our PAF trackers including DiMP_PAF and SiamRPN_PAF with the same baselines on VOT2019. Compared with these state-of-the-arts, DiMP_PAF achieves the top three places on EAO and Robustness, which are 0.8% and 1.9% higher than DiMP. Besides, SiamRPN_PAF outperforms SiamRPN++ on all metrics, which further demonstrates the generalization of our PAF.

3) About Efficiency: Some part-based methods [25], [26] explore offline training for real-time tracking. However, their training often costs massive GPU resources compared to our online-only PAF. The other trackers [63], [64], [65] exploit multiple parts with context regressors for strong robustness, but suffer in real applications due to expensive online computational costs. Compared with them, our PAF adopts fewer part numbers to obtain faster speed with lower costs, thus enabling some PAF trackers (*e.g.*, SiamFC_PAF, TADT_PAF) for real-time tracking. However, our PAF still has limitations in efficiency compared with offline training methods. Intuitively, the sharing strategy of the feature extraction on multiple parts may further help save computational costs. But this would be outside the scope of this work. We leave this to future study.

V. CONCLUSION

In this paper, we propose a generic part-aware framework to improve the robustness of siamese and DCF trackers with context regression, which fully explores not only the global and local parts of the target but also their spatial-temporal relationship to be collaboratively aware of the target state. Our framework adopts context regressors corresponding to multiple parts to keep track of the target online and adaptively aggregate the coarse predictions and the spatial-temporal measures of these regressors to refine the tracking results. The combination window functions are also proposed in part regressors to adaptively filter redundant noise by considering their divergence. Furthermore, the auxiliary scale estimation based on the spatial-temporal outputs of part regressors is also leveraged to boost tracking performance. Numerous evaluations on the popular OTB, TC128, UAV, UAVDT, VOT, TrackingNet, GOT-10k, and LaSOT benchmarks show that our PAF variants based on siamese and DCF whether using the multi-scale search or bounding box regression perform favorably against their baselines and achieve the comparable accuracy with state-of-the-art methods. Immediate future works will include the efficiency optimization of the framework for real applications.

REFERENCES

[1] A. Smeulders, D. Chu, R. Cucchiara, and S. Calderara, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[2] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3516–3527, Jul. 2019.

[3] J. Gao, T. Zhang, and C. Xu, "SMART: Joint sampling and regression for visual tracking," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3923–3935, Aug. 2019.

[4] Z. Li, W. Wei, T. Zhang, M. Wang, S. Hou, and X. Peng, "Online multi-expert learning for visual tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 934–946, 2020.

[5] G. Luo, H. Zhang, H. He, J. Li, and F.-Y. Wang, "Multiagent adversarial collaborative learning via mean-field theory," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4994–5007, Oct. 2021.

[6] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[7] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[8] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.

[9] H. Fan, L. Bai, L. Lin, and F. Yang, "LaSOT: A high-quality large-scale single object tracking benchmark," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 439–461, 2021.

[10] L. Bertinetto and J. Valmadre, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 850–865.

[11] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[12] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6269–6277.

[13] Z. Zhang, Y. Liu, B. Li, W. Hu, and H. Peng, "Toward accurate pixelwise object tracking via attention retrieval," *IEEE Trans. Image Process.*, vol. 30, pp. 8553–8566, 2021.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[15] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[16] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6191.

[17] C. Mayer, M. Danelljan, D. Pani Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13444–13454.

[18] H. Yu, G. Li, and W. Zhang, "The unmanned aerial vehicle benchmark: Object detection, tracking and baseline," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1141–1159, 2020.

[19] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[20] M. Müller, A. Bibi, and S. Giancola, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 300–317.

[21] X. Lu, C. Ma, and B. Ni, "Deep regression tracking with shrinkage loss," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 353–369.

[22] T. Zhang, C. Xu, and M.-H. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 365–378, Feb. 2019.

[23] Y. Zhang, L. Wang, and J. Qi, "Structured Siamese network for real-time visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 351–366.

[24] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2784–2791.

[25] Z. Zhou, W. Pei, X. Li, H. Wang, F. Zheng, and Z. He, "Saliency-associated object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9866–9875.

[26] Z. Liang and J. Shen, "Local semantic Siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 3351–3364, 2020.

[27] W. Ruan et al., "Multi-correlation filters with triangle-structure constraints for object tracking," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1122–1134, May 2019.

[28] P. L. Mazzeo, P. Spagnolo, M. Leo, P. Carcagnì, M. Del Coco, and C. Distante, "Dense descriptor for visual tracking and robust update model strategy," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 8, pp. 3089–3099, Aug. 2020.

[29] M. Kristan, A. Leonardis, and J. Matas, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 3–53.

[30] M. Kristan, J. Matas, and A. Leonardis, "The seventh visual object tracking VOT2019 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct. 2019, pp. 2206–2241.

[31] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 749–765.

[32] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[33] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 459–474.

[34] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4649–4659.

[35] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, and C. Shen, "Graph attention tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9543–9552.

[36] Z. Zhu, Q. Wang, and B. Li, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–117.

[37] S. Cheng and B. Zhong, "Learning to filter: Siamese relation network for robust tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4421–4431.

[38] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.

[39] W. Han, X. Dong, F. S. Khan, L. Shao, and J. Shen, "Learning to fuse asymmetric feature maps in Siamese trackers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16570–16580.

[40] J. Peng and Z. Jiang, "SiamRCR: Reciprocal classification and regression for visual object tracking," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, 2021, pp. 952–958.

[41] N. Wang, W. Zhou, Y. Song, C. Ma, W. Liu, and H. Li, "Unsupervised deep representation learning for real-time tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 400–418, Sep. 2020.

[42] J. Choi, J. Kwon, and K. M. Lee, "Deep meta learning for real-time target-aware visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 911–920.

[43] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[44] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6668–6677.

[45] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu, "Learn to match: Automatic matching network design for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13339–13348.

[46] Y. Chen, J. Xu, J. Yu, Q. Wang, B. Yoo, and J. Han, "AFOD: Adaptive focused discriminative segmentation tracker," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2020, pp. 666–682.

[47] Z. Fu, Q. Liu, Z. Fu, and Y. Wang, "STMTrack: Template-free visual tracking with space-time memory networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13774–13783.

[48] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, "LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15180–15189.

[49] B. Yu, M. Tang, and L. Zheng, "High-performance discriminative tracking with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 9856–9865.

[50] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 38, no. 2, pp. 1401–1409.

[51] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1135–1143.

[52] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11923–11932.

[53] F. Li, X. Wu, W. Zuo, D. Zhang, and L. Zhang, "Remove cosine window from correlation filter-based visual trackers: When and how," *IEEE Trans. Image Process.*, vol. 29, pp. 7045–7060, 2020.

[54] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4844–4853.

[55] S. Li, S. Zhao, B. Cheng, E. Zhao, and J. Chen, "Robust visual tracking via hierarchical particle filter and ensemble deep features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 179–191, Jan. 2020.

[56] L. Zheng, M. Tang, Y. Chen, J. Wang, and H. Lu, "Learning feature embeddings for discriminant model based tracking," 2019, *arXiv:1906.10414*.

[57] M. Danelljan, G. Bhat, and C. Mayer. (2021). *Pytracking: Visual Tracking Library Based on Pytorch*. [Online]. Available: https://github.com/visionml/pytracking

[58] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[59] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939.

[60] M. Danelljan, L. Van Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7183–7192.

[61] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1571–1580.

[62] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4902–4912.

[63] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 353–361.

[64] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4312–4320.

[65] H. Fan and J. Xiang, "Robust visual tracking via local-global correlation filter," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[66] E. Burceanu and M. Leordeanu, "Learning a robust society of tracking parts using co-occurrence constraints," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 162–178.

[67] S. Li, S. Zhao, B. Cheng, and J. Chen, "Noise-aware framework for robust visual tracking," *IEEE Trans. Cybern.*, vol. 52, no. 2, pp. 1179–1192, Feb. 2022.

[68] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial–temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[69] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2891–2900.

[70] Z. Fu, Z. Fu, Q. Liu, W. Cai, and Y. Wang, "SparseTT: Visual tracking with sparse transformers," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1–7.

[71] D. Zhang, Z. Zheng, R. Jia, and M. Li, "Visual tracking via hierarchical deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 4, pp. 3315–3323.

[72] Z. Ma, L. Wang, H. Zhang, W. Lu, and J. Yin, "RPT: Learning point set representation for Siamese visual tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2020, pp. 653–665.

Shengjie Li received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2020, under the supervisor of Prof. Junliang Chen. He is currently a Lecturer with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His current research interests include the Internet of Things Technology and visual object tracking.

Shuai Zhao (Member, IEEE) received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in June 2014, under the supervisor of Prof. Junliang Chen. He is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He is also with the Guangxi Key Laboratory of Cryptography and Information Security. His current research interests include the Internet of Things Technology and service computing.

Bo Cheng (Member, IEEE) received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include network services and intelligence, the Internet of Things Technology, communication software, and distributed computing.

Junliang Chen received the B.S. degree in electrical engineering from Shanghai Jiaotong University, China, in 1955, and the Ph.D. degree in electrical engineering from the Moscow Institute of Radio Engineering, formerly Soviet Russia, in May 1961. Since 1955, he has been working at the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, where he is currently the Chairperson and a Professor of the Research Institute of Networking and Switching Technology. His research interests include communication networks and next generation service creation technology. He was elected as a member of the Chinese Academy of Science in 1991 and the Chinese Academy of Engineering in 1994, for his contributions to fault diagnosis in stored program control exchange. He received the first, second, and third prizes of the National Scientific and Technological Progress Award in 1988, 2004, and 1999, respectively.