Fast Multi-Object Tracking with Feature Pyramid and Region Proposal Networks

Lorenzo Vaquero, Víctor M. Brea, Manuel Mucientes
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela, Santiago de Compostela, Spain
Email: lorenzo.vaquero.otal@usc.es, victor.brea@usc.es, manuel.mucientes@usc.es

***Abstract*—Many computer vision applications require real-time processing speeds, which prevents them from running an object detector on all frames of the sequence. In such circumstances, it is necessary to resort to motion estimation techniques in order to maintain the identity of the targets. This can be carried out by instantiating multiple single object trackers, if there are few targets, or through methods that globally extract the frame features, in order to share computations. The problem with the latter is that they yield features with limited semantic information and detect changes in the scene by performing multi-scale tests, which is inefficient and prone to errors. To solve these problems and provide accurate tracking for multiple objects in real-time, we propose SiamFAST. SiamFAST includes: a feature-pyramid-based region-of-interest extractor that produces quality features for both object exemplars and search areas; a pairwise depthwise region proposal network to compute fast similarities for several dozens of objects; and a multi-object penalization module in order to suppress the effect of distractors. SiamFAST has been validated on three public benchmarks, achieving leading performance against current state-of-the-art trackers.**

I. Introduction

Many computer vision applications seek to extract in-depth knowledge about what occurs in a video, such as anomaly detectors [1] or automatic summarizers [2]. A common feature of most of these systems is that they rely on multiple specific components to gather baseline information from the scene, in order to then use it to reach more elaborated conclusions. One of the most widely adopted components are visual object trackers, since they maintain the identity of the elements of interest over time, observing their evolution in size and location. If there are multiple targets in the scene, this is usually accomplished by running an object detector in each frame, and then associating the detections to generate tracks.

However, if the use case requires real-time feedback, traditional multiple object tracking (MOT) techniques have to be dismissed, as they are computationally very expensive —only 5 methods submitted to the MOT2020-Challenge [3] are able to run in real-time[1], despite having the detections already available. Moreover, all this is further aggravated by the fact that good-performing detectors alone already have difficulties running in real-time —on an NVIDIA TITAN V, EfficientDet-D3 [4] processes HD720 images at 23 fps. Thus, in this type of situations, motion estimation techniques are often relied upon to maintain the identity of the targets between detections.

[1]We qualify a system/module as real-time-capable if it can handle an HD720 video stream at least at 25 fps on an NVIDIA TITAN V or equivalent.

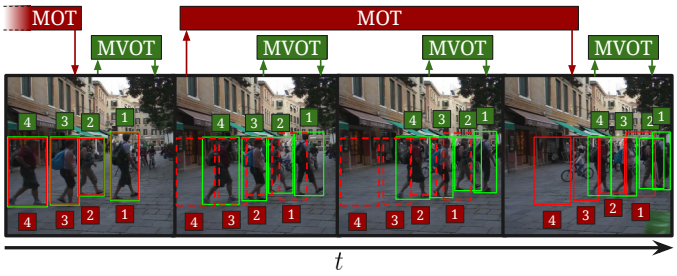

Fig. 1. In a real-time application, an MVOT (green) is able to process the image (upwards arrows) and provide a prediction (downwards arrows) for each frame of the sequence. Meanwhile, an MOT system lacking motion estimation (red) is computationally very expensive, so it is not able to process some frames —it resorts to populate them with the last known position of the objects— and its outputs present noticeable delays. Benchmarks such as VOTChallenge's VOT-RT [5] simulate these scenarios.

Motion estimation mechanisms are able to provide the coordinates of objects in all the frames without depending on continuous detections. Nowadays they are mostly based on Visual Object Tracking (VOT), leading to the concept of Multiple Visual Object Tracking (MVOT) techniques (Fig. 1), which allows them to exploit the most recent breakthroughs in single-object tracking [6]. Moreover, since most of them support arbitrary objects (they are class-agnostic), they can be applied to a wide variety of scenarios without requiring any further retraining or adaptation. However, the main disadvantage they present is that they often resort to the multiple instantiation of single-object trackers, making them viable only for a few targets —as their speed decreases exponentially with each added object. To overcome this problem, techniques such as [7] emerge, which share computations between objects and introduce specialized efficient operators, being able to track several dozens of objects in real-time. Nonetheless, the resulting features are not always of high quality, leading to inferior accuracies.

In this paper we propose SiamFAST (**Siam**ese **F**eature pyr**A**mid propo**S**al **T**racker), a multiple visual object tracker for motion estimation. SiamFAST takes inspiration from the techniques employed in computer vision to generate quality predictions and applies them to tracking, following the basics defined in [7], all while solving several problems of the previous framework. The main contributions of our work can be summarized as follows:

- We enable the tracking of widely varying object sizes

by integrating a Feature Pyramid Network (FPN) into the regions-of-interest (RoI) crop-and-resize operation. It allows for less abrupt scale variations and features with richer semantic information. To the best of our knowledge, this is the first time that such a technique is employed in a detection-independent object tracker.
- We introduce a novel Pairwise Depthwise Region Proposal Network (PD-RPN) that efficiently identifies variations in the objects' aspect ratio and size without resorting to multi-scale testing. This component merges the features of exemplars and search areas in pairs, generating tensors on which anchors are then applied to more accurately predict the coordinates of the objects.
- We introduce a novel multi-object penalization module for a computationally efficient refinement of proposals. It embeds a morphological operator for filtering outliers and a distractor suppression mechanism, which prove especially useful in crowded scenarios.
- We assess our architecture on three public benchmarks using VOT-RT metrics [5], outperforming current state-of-the-art trackers.

II. Related work

A. Visual Object Tracking

Visual object trackers define a function that compares the appearance of an object at the beginning of a video —exemplar— with the current frame or a portion of it —search area—. This comparison will yield the new coordinates of the object in the scene. Most single object trackers provide predictions as they process frames —online tracking— and do not vary the exemplar image of the object, which makes them efficient, and good potential candidates for real-time applications.

Traditionally, the similarity between the exemplar and the search areas was modeled through Discriminative Correlation Filters (DCF). Initially, these selected the single filter that resulted in the highest correlation with the exemplar, and then applied it over the following search areas, thus being able to distinguish the background from the target [8]. Given the effectiveness and simplicity of the technique, it was gradually developed to incorporate more types of filters, even modeling them through convolutional learned layers [9]. However, while DCFs are fast, they have now been displaced in favor of more accurate deep learning-based trackers.

The current state-of-the-art in visual object tracking relies on deep learning techniques. They are based on deep convolutional neural networks (CNNs) that are trained offline over large databases, which produces accurate and robust similarity functions. These networks extract the features of the exemplar and fuse them with those of the search area, generating predictions. This architecture can be modeled with a Siamese structure with shared weights, which is very efficient and helps learning [10]. On this basis, new contributions from other fields of computer vision were incorporated, such as the inclusion of region proposal networks —with [11] or without anchors [12]—, segmentation heads [13], deeper backbones [14], or mechanisms to assess the quality of the predictions [15].

Besides Siamese networks, there are also convolutional architectures that feature different components for estimating and classifying objects [16]. These trackers incorporate an estimation module to predict the overlap between the output bounding-box and the object, and a classification component to discriminate the object from the background and possible distractors. The results they deliver are generally good, but they are strongly affected by the initialization of the exemplar —as they are not deterministic— and are very sensitive to the chosen hyperparameters. This is why specific mechanisms have been proposed to reinitialize the starting appearance model until its quality is satisfactory [17]. Nonetheless, the nondeterminism of these architectures together with the high computational cost of refining initializations and predictions makes them unsuitable as real-time MVOT systems.

B. Motion estimation

The most widely used methods for tracking multiple objects in a video rely on the association of detections. To do so, they are highly dependent on a quality detector that must run on all frames and result in few false positives while providing good recall [18]. Since the most accurate detectors are very resource-expensive, this approach is only feasible when no constraint is imposed on the system speed. If the objective is to track multiple objects in real-time, the paradigm changes, and motion estimation is required for those periods when no detections are available.

Currently, the preferred method for estimating the motion of multiple targets in real-time video is based on the use of multiple visual object trackers (MVOT) [6]. As the name implies, they employ models that rely on visual features to locate objects from one frame to the next. Given their flexibility and efficiency, single-object visual trackers are often employed to accomplish this task, either as part of some modules of the architecture [19] or by incorporating them as a completely independent component [20]. However, the problem with these approaches is that they instantiate one tracker per object, resulting in highly inconsistent performance and being only usable for environments with few objects.

To solve the aforementioned problem, it is necessary to develop MVOT architectures capable of scaling naturally with the number of objects. Thus, approaches such as [7] arise, which introduces new specialized operators and shares most of the computations between targets, having a practically constant computational cost, regardless of the number of objects. This allows it to track several dozen objects in real time, being much more efficient than other approaches — [7] handles 100 objects in an FHD video at 25 fps while [19] runs at 5 fps for just 21 targets. However, the techniques it employs present some problems —it is not able to track very large or very small objects without changing the input frame size— and the outputs it provides are error prone —it resorts to multi-scale testing and cannot recognize scale changes.

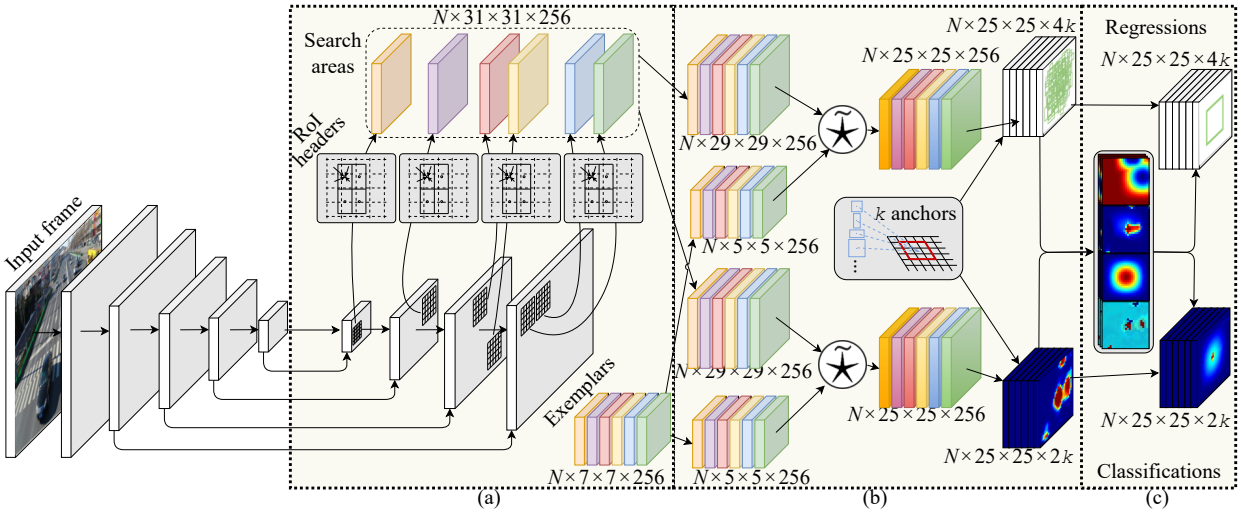

Fig. 2. SiamFAST's architecture. The (a) region-of-interest extractor obtains high-quality features for exemplars and search areas. The (b) pairwise-depthwise region proposal network performs efficient comparisons, providing the new coordinates for each object. The (c) multi-object penalization module refines the predictions to make them more robust against distractors and outliers.

III. SiamFAST Network Architecture

SiamFAST has been designed with the aim of sharing as many computations between objects as possible, and using specialized operators that provide improved accuracy while naturally scaling to several dozens of targets. As depicted in Fig. 2, SiamFAST first extracts the global frame features, to then isolate search areas using a region-of-interest (RoI) extractor that embeds a feature pyramid network (FPN) (a). Following this, exemplar and search area features are compared 2-by-2 using a pairwise depthwise region proposal network (PD-RPN) (b). This yields maps with objectness classifications and bounding-box regressions that are later refined using a multi-object penalization module (c). The most relevant components are explained in more detail below.

A. Region-of-Interest Extractor

SiamFAST computes the updated coordinates of the targets in each new frame. To do this, it could compare the exemplar appearance model of each object with the whole frame. However, this would be computationally very expensive and would have problems finding the objects if they had changed in size. This is why visual object trackers rely on the use of search areas, which are small portions of the scene that have been cropped to contain the object of interest with an approximately constant size. Trackers such as [11] or [14] perform this cropping on the input image, and then extract their features. This allows them to always work at an optimal resolution but degrades the speed of the system, as the backbone is the most expensive operation and its execution per-region is impractical. On the other hand, [7] first extracts the features of the entire frame —sharing computations between objects—, and then crops and resizes the deepest feature map to create the search areas. This allows it to be extremely efficient, but the quality of the features suffers greatly if the objects do not have a particular size —between 100 px^2 and 200 px^2.

To overcome these problems, SiamFAST shares computations by running the backbone on the entire frame, and then generates high-quality search areas using the region-of-interest extractor shown in Fig. 2a. Thus, SiamFAST does not rely only on the last layer of the backbone, but integrates its intermediate levels through a feature pyramid network that generates semantically rich features at different resolutions. Forward propagation of the deeper layers pushes the higher-level information to the rest of the network, while connections with shallower layers allows retaining object details that are lost as the resolution decreases. This architecture will therefore provide 4 levels of enriched features whose resolutions will allow to faithfully represent objects between 64 px^2 and 512 px^2. Unlike object detectors that employ similar techniques [21], we consider FPN level 1 since the scene will usually contain dozens of not very large objects, and do not want to generate very deep features —levels 5 or 6— as they are too abstract and not good for discriminating between very similar objects. To the best of our knowledge, this is the first time that this technique is exploited in a detection-independent object tracker.

For each resolution level, SiamFAST has a header that performs the remaining region-of-interest extraction process. This header first applies a 1×1 convolution on each enriched feature map. The goal of this operation is to transform the features to a similar feature representation, in order to make them comparable regardless of the original resolution they came from. This allows the rest of the network architecture —i.e., similarity operation and multi-object penalization module— to be shared for all levels, which eases the convergence of the algorithm and improves the efficiency. The latter is highly significant, as it allows SiamFAST to compare exemplars and search areas originally extracted at different resolutions, something that other visual object trackers do not allow —e.g., SiamRPN++ [14] and SiamCAR [12] have to modify the stride in their backbones to ensure that all comparisons are performed at the same resolution, and employ an ensemble of heads for the different levels of the network.

Finally, each header clips the regions corresponding to each search area through a modified RoI Align [22], stacking them all in the same tensor. This operation has one sampling point per bin, for a better computational efficiency, and ensures that regions are always rendered with the same aspect ratio in relation to the frame (1:1), which facilitates subsequent similarity functions. The decision as to whether the header at level $k \in [1, 4]$ will manage an area B with coordinates (B^x, B^y, B^w, B^h) is calculated as follows:

$$\begin{cases} \text{True}, & \text{if } k = \min\left(k_{lo}, \max\left(k_{hi}, \left\lfloor k_0 + \log_2\left(\frac{\sqrt{B_w B_h}}{s\lfloor a/s \rfloor}\right)\right\rfloor\right)\right) \\ \text{False}, & \text{otherwise} \end{cases} \quad (1)$$

where a and s are the canonical sizes of the search area and network stride in Siamese trackers, respectively —$a = 255$ and $s = 8$ for SiamFC-based architectures—, k_{hi} is the level with the highest resolution —i.e., 4—, k_{lo} is the level with the lowest resolution —i.e., 1—, and k_0 is the level with resolution $1/s$ —level 3 in our case. Lastly, the crop coordinates for areas in layer k are calculated as $B/2^k$.

B. Similarity Operation

The next step in the pipeline of a Siamese tracker involves comparing the exemplars with the search areas in a sliding-window manner, in order to obtain the updated coordinates of the objects. This can be carried out with a simple cross-correlation that directly returns a flat score map per object with its probabilities of occurrence in each location of the search area [7], [10]. It is extremely efficient and is very effective in finding the center of the objects —as it is the only thing that the similarity function learns—, but it is unable to detect size variations. This is why it is necessary to combine it with multi-scale testing —which therefore requires a total of $\mathcal{S} * \mathcal{N}$ comparisons, where $\mathcal{S}$ is the number of considered scales and $\mathcal{N}$ is the amount of targets. Nonetheless, this makes it more prone to instabilities and does not allow handling variations in the aspect ratio of the objects, so some approaches emerge trying to mitigate this by posing the similarity operation as a region proposal network (RPN) [11].

An RPN is based on the use of anchors, which are rectangles of different sizes and aspect ratios that slide across the search area. Thus, when applied to tracking, the objective of the network is to predict the probability of each anchor containing the target and to redefine its location and size to fit said object. This is implemented through 2 branches: one for the classification of an anchor between object and non-object —objectness— and another for the regression of the bounding box —coordinates—. This removes the need for feeding multiple search areas at different scales, as the regressor takes care of detecting size and aspect ratio changes. However, to apply the RPN, [11] first transforms the features of exemplars and search areas depending on the number of anchors K to then cross-correlate them, resulting in $6K\mathcal{N}$ comparisons —$4K\mathcal{N}$ for the coordinates regression and $2K\mathcal{N}$ for the objectness classification—. This has a very large impact on the speed of the system as the number of objects grows. This is why, although SiamFAST's similarity operation is based on this concept, it also takes the idea described in [14] of postponing the anchor-dependent transformations.

SiamFAST implements its similarity operation through the pairwise-depthwise region proposal network (PD-RPN) described in Fig. 2b. First, the tensors containing the exemplars and the search areas go through a 3×3 convolution that focuses on object/background distinction and another 3×3 convolution that specializes them for area delimitation, thus creating the root of the two branches of the RPN. At this point, the exemplar tensors can be cached and reused during the rest of the tracking process, in order to speed up the computations. Following this, within each branch, the feature tensors are merged together. This is accomplished using a combination of reshapes and the pairwise cross-correlation ($\tilde{\star}$) defined in [7]. Thus, the features are fused depthwise —without aggregating their channels, which keeps their information highly discriminative— and in a very efficient manner, since $\tilde{\star}$ is designed to have a practically constant computational cost regardless of the number of objects. Finally, anchors are applied on the resulting tensors, yielding the classification and regression predictions for each object. This entire process requires only $2\mathcal{N}$ lightweight comparisons, involving an order of magnitude fewer parameters than a conventional RPN, which allows SiamFAST to combine speed and accuracy.

C. Multi-Object Penalization Module

Although the predictions generated by the PD-RPN are mostly sound, they can be improved through heuristic knowledge, as most visual object trackers do [10]–[12], [15], [17]. The preferred and most effective methods are based on establishing a lower base probability to large displacements and to bounding box transformations that involve large variations in the area or the aspect ratio. These give good results, however, since SiamFAST operates in environments with potentially several dozens of objects, there is still room for improvement to further refine its predictions. Thus, as shown in Fig. 2c, SiamFAST introduces a multi-object penalization module that applies the two previously stated techniques plus two novel ones: a morphological penalization and a distractor penalization, taking advantage of the fact that it will have information from many of the objects in the scene. The proposed new techniques are detailed below.

Morphological penalization. Occasionally, isolated points with excessively high probabilities appear in the classification score map. These outliers are a problem, since they are not associated with anything specific —it can be anything from a stuck pixel in the video to a complex background with unusual colors— and represent failures that overshadow the real predictions of the network and prevent it from working properly. To solve this problem, we propose a novel penalization based on the morphological erosion operation, which is able to reduce the shapes contained in grayscale images. Thus, we slide a 3×3 kernel per channel —since the activations for different anchor types are weakly related— that suppresses such outliers and spares the large high-probability areas, which are mostly related to correct predictions.

This process has the added effect of shrinking the correct matches, causing them to cover a smaller area of the object. This can be reversed by applying a 3×3 morphological dilation kernel after the erosion. However, according to our experiments, tightening the predictions of the network in crowded scenarios has a positive impact on its performance. This is why in such situations we only apply erosion.

Distractor-aware penalization. In videos with multiple targets, it is common for those elements to be similar to each other —e.g., of the same caterogry—, so identity switches are a concern. Since SiamFAST will maintain the identity of several objects at once, this allows it to have a strong knowledge regarding what occurs in the scene and, thus, consider such elements as potential distractors. Hence, we propose a novel lightweight penalization mechanism capable of modeling the interactions between objects in the video in a simple-yet-effective manner.

The distractor model is rendered as a probabilistic map $G_{W\times H\times\mathcal{N}}$ that is proportional to the size of the scene —$(W,H) = \left\lceil \frac{M*F}{a} \right\rceil$ —where M is the size of the PD-RPN classification map, F is the dimensions of the frame, and a is the canonical search area size of Siamese trackers [10]. Each channel represents one of the $\mathcal{N}$ objects and captures its contribution to the distractor model in the following way:

$$\begin{cases} 1 - \left(\sin\left(\pi \frac{A_p^x - 0.5A_p^w - i}{A_p^w - 1}\right) \sin\left(\pi \frac{A_p^y - 0.5A_p^h - j}{A_p^y - 1}\right)\right)^2, \\ \quad \text{if } i \in \left[A_p^x - \frac{A_p^w}{2}, A_p^x + \frac{A_p^w}{2}\right] \text{ and } j \in \left[A_p^y - \frac{A_p^h}{2}, A_p^y + \frac{A_p^h}{2}\right] \\ 1, \quad \text{otherwise} \end{cases} \tag{2}$$

where $i \in [0,W)$, $j \in [0,H)$, $p \in [0,\mathcal{N})$, and $A_{4\times\mathcal{N}} = \left\{(A_p^x, A_p^y, A_p^w, A_p^h)\right\}_{p\in[0,\mathcal{N})}$ is the search area coordinates of the objects mapped over the probabilistic map. Thus, each object will generate a low-probability region centered on its location and proportional to its size, making it harder to consider network proposals in that area.

Once the global probabilistic model G is available, it must be adapted to each specific object. Thus, a local penalization tensor $L_{M_w\times M_h\times\mathcal{N}}$ is created, which contains the windows to be applied to the classification predictions of each object. This tensor is computed as follows:

$$L_p = \min_{d\in\mathcal{D}} \kappa_M\left(G_d, A_p\right) \tag{3}$$

where $\mathcal{D} := \{\forall d \in [0,\mathcal{N}), d \neq p\}$ and κ_M is a crop-and-resize operator —e.g., RoI Align [22]— with an output of size M. At this point, L can now be applied to the output of the classification branch to suppress potential proposals biased by distractors. This whole process is very efficient —G needs to be computed only once per frame— and delivers good results, which makes it very well suited for MVOT solutions.

IV. Experiments

We conducted experiments in different multi-object scenarios in order to verify the performance of SiamFAST. The experiments were run using TensorFlow on a computer with an NVIDIA TITAN Xp and an Intel Core i7-9700K with 16 GB of DDR4 RAM.

A. Implementation details

Feature extractor. It is important to employ a backbone with padding in its convolutions, so the information at the edges of the image is not diluted. Thus, we chose ResNet-18 [23] as it also provides robust features without requiring too much memory or computational overhead. Since we aim to capture the details of the scene at different resolutions, we maintain the stride of all the convolutional blocks of the backbone —unlike other trackers [12], [14] that modify them to have unit spatial stride. Lastly, we do not adopt deeper versions of ResNet since the gain in feature quality they provide does not compensate for the loss of speed.

Exemplar and search area sizes. Our PD-RPN produces outputs of size 25×25 from exemplars of size 7×7 and search areas of size 31×31, as [14] shows that they provide good accuracy without consuming excessive computational resources. The area A^2 covered by an exemplar is determined by the size (w, h) of the object:

$$A^2 = \left(w + \frac{1}{2}(w+h)\right) \times \left(h + \frac{1}{2}(w+h)\right) \tag{4}$$

This is mapped to 15×15 bins, and then the central 7×7 region is extracted —as we need no additional context. Regarding the search area, it also depends on the size of the target, and covers an area of size $\left(\frac{31A}{15}\right)^2$, which is mapped to 31×31 bins.

Training process. SiamFAST is trained on ILSVRC [24], YT-BB [25], GOT-10k [26], and COCO [27], following the spatial-aware sampling strategy described in [14] and ensuring that the FPN receives objects of all sizes throughout the whole process. We use a batch size of 16 and optimize the regression and classification losses defined in [11] using an Adam optimizer [28] that starts from a learning rate of 3×10^{-5} and is exponentially decayed to 3×10^{-7}. The backbone is pre-trained on ImageNet and is frozen during the first 15 epochs, to then end-to-end train the network for 30 additional epochs. Finally, we fine-tune SiamFAST during 15 epochs in which we favor intra-class discrimination and samples extracted from videos. We consider 5 anchors with ratios $[\frac{1}{3}, \frac{1}{2}, 1, 2, 3]$.

B. Comparison with the state-of-the-art

The role of an MVOT is to receive the coordinates of several objects and provide their subsequent locations in real-time, without having access to further detection information. This is why these systems must present a good balance between speed and tracking quality. Therefore, we employed VOTChallenge's VOT-RT metrics [5] to evaluate the performance of SiamFAST and other state-of-the-art approaches. This benchmark provides the robustness —percentage of frames where the tracker did not lose the object, with an exponential sensitivity of $\mathcal{S} = 30$— and accuracy —average overlap between ground truths and predictions— for different speed thresholds —i.e., 25 fps and 20 fps.

In order to have a representative set of scenarios in which motion estimation systems are commonly used, we have selected various multi-object public datasets: MOT-2017 [29], UAVDT [30], and VisDrone [31]. We compared SiamFAST against the leading state-of-the-art

TABLE I
VOT-RT METRICS RESULTS

	MOT-2017				UAVDT				VisDrone			
	@20 fps		@25 fps		@20 fps		@25 fps		@20 fps		@25 fps	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SiamMT	*54.5*	*76.2*	*54.5*	*73.8*	*54.6*	*92.3*	*54.5*	*92.3*	*45.7*	*52.2*	*45.4*	*51.5*
SiamFC++	53.9	70.2	52.3	67.6	52.4	86.1	48.5	81.2	34.2	32.8	32.0	31.5
SiamCAR	47.5	60.0	46.0	58.2	40.3	70.0	37.1	64.9	28.9	29.8	27.8	29.3
SiamRPN++	48.5	60.1	47.0	58.9	38.2	67.1	35.2	62.5	28.2	29.6	27.2	29.1
SiamRPN	50.9	70.7	49.5	67.9	53.5	89.9	49.8	85.7	36.2	34.4	33.9	32.8
SiamFC	45.8	58.4	45.2	57.2	41.4	73.3	37.5	67.3	27.9	29.5	27.0	29.0
SiamFAST	**57.7**	**77.5**	**57.7**	**76.6**	**57.5**	**93.3**	**57.4**	**93.4**	**50.5**	**59.2**	**49.1**	**57.6**

TABLE II
ABLATION STUDY WITH VOT-RT METRICS @25 FPS

	MOT-2017		UAVDT		VisDrone	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
Baseline	49.5	74.1	41.9	75.9	42.3	48.7
+ R	55.2	76.3	50.9	92.5	44.8	54.4
+ R + S	56.7	76.2	57.0	93.4	47.0	56.3
+ R + S + P	57.7	76.6	57.4	93.4	49.1	57.6

methods SiamFC [10], SiamRPN [11], SiamRPN++ [14], SiamCAR [12], SiamFC++ [15], and SiamMT [7], which are frequently used as MVOTs for motion estimation tasks. The results for each algorithm are shown in Table I. **Red**, *blue* and green, represent 1st, 2nd and 3rd respectively.

As shown, SiamFAST improves the state-of-the-art in the tested datasets by generous margins. It outperforms the best state-of-the-art algorithm —SiamMT— by an average of +3.5 points in accuracy and +3.2 points in robustness —computed for both 20 fps and 25 fps—, and beats all other trackers by more than +8.4 and +12.7 points in average accuracy and robustness, respectively. This demonstrates the effectiveness and versatility of SiamFAST and its novel components, being able to operate in very different and varied environments without any additional tuning.

MOT-2017 contains several pedestrian videos under a variety of settings: both indoors and outdoors with different lighting conditions. These sequences are recorded with both overhead fixed cameras and low-angle mobile cameras, which increase the amount of prolonged total occlusions. SiamFAST performs very well in these situations thanks to its RoI extractor (+3.2 points in accuracy and +2.8 points in robustness @25 fps when compared to SiamMT), generating semantically similar features capable of describing an object no matter if it is far away or covers a large part of the scene. Additionally our multi-object penalization module is able to resolve several uncertainties found in the displayed crowded environments.

UAVDT features numerous intersections and highways from a zenithal point of view, showing a huge variety of vehicles with few occlusions but plenty of small objects. Thanks to SiamFAST's PD-RPN we can vary the aspect ratio of the bounding-boxes —something very necessary in this kind of aerial videos—, which results in a great improvement of the tracking quality, especially in accuracy (+2.9 and +1.1 points in accuracy and robustness @25 fps when compared to the best state-of-the-art approach). Furthermore, our RoI extractor obtains meaningful features from the smallest objects without having to resize the images fed to the network —unlike SiamMT or SiamRPN do, for example.

VisDrone is composed of multiple videos captured with drone-mounted cameras in urban and city environments, covering many different weather conditions both during day and during night. The camera is in motion, the categories captured are very different —e.g., cars, awning-tricycles, people, bicycles— and occlusions are very common. To minimize identity switches and increase the robustness, the multi-object penalization module becomes critical in this type of crowded environments. This, coupled with the fact that the videos depict a large number of objects with very different sizes —our RoI extractor is able to correctly extract their features— and that perspective changes are very frequent —our PD-RPN is able to efficiently vary the aspect ratio of the bounding-boxes—, gives SiamFAST the biggest advantage in this benchmark over the other approaches (we outperform the state-of-the-art by +3.7 points in accuracy and +6.1 points in robustness @25 fps).

C. Ablation Study

We assess the contribution of each component in Table II. **Baseline** is a version of SiamMT [7] with a ResNet-18 [23] as the backbone. It lags behind SiamMT as it does not dynamically modify the size of the frame. **R** is the region-of-interest extractor (Section III-A), which increases the accuracy and robustness by an average of +5.7 and +8.2 points, respectively. **S** is the similarity operation (Section III-B), whose better bounding boxes yield an average improvement of +3.3 and +0.9 points in accuracy and robustness, respectively. **P** is the multi-object penalization module (Section III-C) and increases accuracy by an average of +1.2 points and robustness by an average of +0.6 points. Overall, the proposed novelties provide an improvement over the baseline of +10.2 and +9.6 points in average accuracy and robustness, respectively. An extended ablation study can be found in the supplementary material.

V. CONCLUSIONS

We have presented a new real-time MVOT (multiple visual object tracker) named SiamFAST. The main novelties it introduces are an RoI extractor that generates quality features, a PD-RPN that efficiently produces bounding-boxes for several dozens of objects and a multi-object penalization module that refines predictions in crowded scenarios. We have assessed the performance of SiamFAST in different real-time benchmarks under very different conditions, outperforming the best state-of-the-art tracker by an average of +3.3 points in both average accuracy and robustness at 25 fps.

ACKNOWLEDGMENT

This research was partially funded by the Spanish Ministerio de Ciencia e Innovación [grant numbers PID2020-112623GB-I00, RTI2018-097088-B-C32], and the Galician Consellería de Cultura, Educación e Universidade [ED431C 2018/29, ED431C 2021/048, ED431G 2019/04]. Grants are co-funded by the European Regional Development Fund. Lorenzo Vaquero is supported by Spanish Ministerio de Universidades under the FPU national plan (FPU18/03174).

REFERENCES

[1] G. Orrù, D. Ghiani, M. Pintor, G. L. Marcialis, and F. Roli, "Detecting anomalies from video-sequences: a novel descriptor," in *IEEE Int. Conf. Pattern Recognit. (ICPR)*, 2020, pp. 4642–4649.

[2] O. Elharrouss, N. Almaadeed, S. Al-Máadeed, A. Bouridane, and A. Beghdadi, "A combined multiple action recognition and summarization for surveillance video sequences," *Appl. Intell.*, vol. 51, pp. 690–712, 2021.

[3] P. Dendorfer *et al.*, "MOT20: A benchmark for multi object tracking in crowded scenes," *CoRR*, vol. abs/2003.09003, 2020.

[4] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10 778–10 787.

[5] M. Kristan *et al.*, "The seventh visual object tracking VOT2019 challenge results," in *IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, 2019, pp. 2206–2241.

[6] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[7] L. Vaquero, M. Mucientes, and V. M. Brea, "Tracking more than 100 arbitrary objects at 25 fps through deep learning," *Pattern Recognit.*, vol. 121, p. 108205, 2022.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 2544–2550.

[9] D. Yuan, X. Li, Z. He, Q. Liu, and S. Lu, "Visual object tracking with adaptive structural convolutional network," *Knowl. Based Syst.*, vol. 194, p. 105554, 2020.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conf. Comput. Vis. (ECCV) Workshops*, 2016, pp. 850–865.

[11] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8971–8980.

[12] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "Siamcar: Siamese fully convolutional classification and regression for visual tracking," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 6268–6276.

[13] Y. Yin, D. Xu, X. Wang, and L. Zhang, "Agunet: Annotation-guided u-net for fast one-shot video object segmentation," *Pattern Recognit.*, vol. 110, p. 107580, 2021.

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

[15] Y. Xu, Z. Wang, Z. Li, Y. Ye, and G. Yu, "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 12 549–12 556.

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

[17] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6181–6190.

[18] M. Fernández-Sanjurjo, M. Mucientes, and V. Brea, "Real-time multiple object visual tracking for embedded GPU systems," *IEEE Internet Things J.*, vol. 8, pp. 9177–9188, 2021.

[19] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 6767–6776.

[20] Z. Zhou, W. Luo, Q. Wang, J. Xing, and W. Hu, "Distractor-aware discrimination learning for online multiple object tracking," *Pattern Recognit.*, vol. 107, p. 107512, 2020.

[21] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 936–944.

[22] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[24] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7464–7473.

[26] 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.*, pp. 1–1, 2019.

[27] T. Lin *et al.*, "Microsoft COCO: common objects in context," in *European Conf. Comput. Vis. (ECCV) Workshops*, 2014, pp. 740–755.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Repr. (ICLR)*, 2015, pp. 1–15.

[29] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016.

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

[31] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," *CoRR*, vol. abs/2001.06303, 2020.