The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking

Dawei Du[1[0000−0001−9404−524X]], Yuankai Qi[2[0000−0003−4312−5682]], Hongyang Yu[2[0000−0003−0036−531X]], Yifan Yang[1[0000−0003−1455−2001]], Kaiwen Duan[1[0000−0002−8663−7429]], Guorong Li[1[0000−0003−3954−2387]], Weigang Zhang[3[0000−0003−0042−7074]], Qingming Huang[1[0000−0001−7542−296X]], and Qi Tian[4,5[0000−0002−7252−5047]]

[1] University of Chinese Academy of Sciences, China
{dawei.du,yifan.yang,kaiwen.duan}@vipl.ict.ac.cn,
liguorong@ucas.ac.cn,qmhuang@ucas.ac.cn
[2] Harbin Institute of Technology, China
qykshr@gmail.com, hyang.yu@hit.edu.cn
[3] Harbin Institute of Technology, Weihai, China
wgzhang@hit.edu.cn
[4] Huawei Noah's Ark Lab, China
tian.qi1@huawei.com
[5] University of Texas at San Antonio, USA
qi.tian@utsa.edu

Abstract. With the advantage of high mobility, Unmanned Aerial Vehicles (UAVs) are used to fuel numerous important applications in computer vision, delivering more efficiency and convenience than surveillance cameras with fixed camera angle, scale and view. However, very limited UAV datasets are proposed, and they focus only on a specific task such as visual tracking or object detection in relatively constrained scenarios. Consequently, it is of great importance to develop an unconstrained UAV benchmark to boost related researches. In this paper, we construct a new UAV benchmark focusing on complex scenarios with new level challenges. Selected from 10 hours raw videos, about 80,000 representative frames are fully annotated with bounding boxes as well as up to 14 kinds of attributes (*e.g.*, weather condition, flying altitude, camera view, vehicle category, and occlusion) for three fundamental computer vision tasks: object detection, single object tracking, and multiple object tracking. Then, a detailed quantitative study is performed using most recent state-of-the-art algorithms for each task. Experimental results show that the current state-of-the-art methods perform relative worse on our dataset, due to the new challenges appeared in UAV based real scenes, *e.g.*, high density, small object, and camera motion. To our knowledge, our work is the first time to explore such issues in unconstrained scenes comprehensively. The dataset and all the experimental results are available in https://sites.google.com/site/daviddo0323/.

Keywords: UAV, Object Detection, Single Object Tracking, Multiple Object Tracking

1 Introduction

With the rapid development of artificial intelligence, higher request to efficient and effective intelligent vision systems is putting forward. To tackle with higher semantic tasks in computer vision, such as object recognition, behaviour analysis and motion analysis, researchers have developed numerous fundamental detection and tracking algorithms for the past decades.

To evaluate these algorithms fairly, the community has developed plenty of datasets including detection datasets (*e.g.*, Caltech [14] and DETRAC [46]) and tracking datasets (*e.g.*, KITTI-T [19] and VOT2016 [15]). The common shortcoming of these datasets is that videos are captured by fixed or moving car based cameras, which is limited in viewing angles in surveillance scene.

Benefiting from flourishing global drone industry, Unmanned Aerial Vehicle (UAV) has been applied in many areas such as security and surveillance, search and rescue, and sports analysis. Different from traditional surveillance cameras, UAV with moving camera has several advantages inherently, such as easy to deploy, high mobility, large view scope, and uniform scale. Thus it brings new challenges to existing detection and tracking technologies, such as:

- **High Density.** Since UAV cameras are flexible to capture videos at wider view angle than fixed cameras, leading to large object number.
- **Small Object.** Objects are usually small or tiny due to high altitude of UAV views, resulting in difficulties to detect and track them.
- **Camera Motion.** Objects move very fast or rotate drastically due to the high-speed flying or camera rotation of UAVs.
- **Realtime Issues.** The algorithms should consider realtime issues and maintain high accuracy on embedded UAV platforms for practical application.

To study these problems, limited UAV datasets are collected such as Campus [39] and CARPK [22]. However, they only focus on a specific task such as visual tracking or detection in constrained scenes, for instance, campus or parking lots. The community needs a more comprehensive UAV benchmark in unconstrained scenarios for further boosting research on related tasks.

To this end, we construct a large scale challenging UAV Detection and Tracking (UAVDT) benchmark (*i.e.*, about $80,000$ representative frames from 10 hours raw videos) for 3 important fundamental tasks, *i.e.*, object DETection (DET), Single Object Tracking (SOT) and Multiple Object Tracking (MOT). Our dataset is captured by UAVs[6] in various complex scenarios. Since the current majority of datasets focus on pedestrians, as a supplement, the objects of interest in our benchmark are *vehicles*. Moreover, these frames are manually annotated with bounding boxes and some useful attributes, *e.g.*, vehicle category and occlusion. This paper makes the following contributions: (1) We collect a fully annotated dataset for 3 fundamental tasks applied in UAV surveillance. (2) We provide an extensive evaluation of the most recently state-of-the-art algorithms in various attributes for each task.

[6] We use DJI Inspire 2 to collect videos, and more information about the UAV platform can be found in http://www.dji.com/inspire-2.

2 UAVDTBenchmark

The UAVDTbenchmark consists of 100 video sequences, which are selected from over 10 hours of videos taken with an UAV platform at a number of locations in urban areas, representing various common scenes including squares, arterial streets, toll stations, highways, crossings and T-junctions. The average, min, max length of a sequence are 778.69, 83 and $2,970$ respectively. The videos are recorded at 30 frames per seconds (fps), with the resolution of 1080×540 pixels.

Table 1. Summary of existing datasets ($1k = 10^3$). D=DET, M=MOT, S=SOT.

Datasets	Attributes									
	UAV	Frames	Boxes	Tasks	Vehicles	Weather	Occlusion	Altitude	View	Year
MIT-Car [34]		$1.1k$	$1.1k$	D	✓					2000
Caltech [14]		$132k$	$347k$	D			✓			2012
KAIST [23]		$95k$	$86k$	D		✓	✓			2015
KITTI-D [19]		$15k$	$80.3k$	D	✓		✓			2014
MOT17Det [1]		$11.2k$	$392.8k$	D			✓			2017
CARPK [22]	✓	$1.5k$	$90k$	D	✓					2017
Okutama [3]	✓	$77.4k$	$422.1k$	D						2017
PETS2009 [18]		$1.5k$	$18.5k$	D,M		✓				2009
KITTI-T [19]		$19k$	$> 47.3k$	M	✓		✓			2014
MOT15 [26]		$11.3k$	$> 101k$	M		✓				2015
DukeMTMC [38]		$2852.2k$	$4077.1k$	M			✓			2016
DETRAC [46]		$140k$	$1210k$	D,M	✓	✓	✓			2016
Campus [39]	✓	$929.5k$	$19.5k$	M	✓					2016
MOT16 [29]		$11.2k$	$> 292k$	M		✓	✓			2016
MOT17 [1]		$11.2k$	$392.8k$	M		✓	✓			2017
ALOV300 [40]		$151.6k$	$151.6k$	S						2015
OTB100 [49]		$59k$	$59k$	S						2015
VOT2016 [15]		$21.5k$	$21.5k$	S			✓			2016
UAV123 [31]	✓	$110k$	$110k$	S	✓					2016
UAVDT	✓	$80k$	$841.5k$	D,M,S	✓	✓	✓	✓	✓	2018

2.1 Data Annotation

For annotation, we ask over 10 domain experts to label our dataset using the *vatic* tool[7] for two months. With several rounds of double-check, the annotation errors are reduced as much as possible. Specifically, about $80,000$ frames in the UAVDTbenchmark dataset are annotated over $2,700$ vehicles with 0.84 million bounding boxes. According to PASCAL VOC [16], the regions that cover too small vehicles are ignored in each frame due to low resolution. Figure 1 shows some sample frames with annotated attributes in the dataset.

Based on different shooting conditions of UAVs, we first define 3 attributes for MOT task:

– **Weather Condition** indicates illumination when capturing videos, which affects appearance representation of objects. It includes *daylight*, *night* and

[7] http://carlvondrick.com/vatic/

Fig. 1. Examples of annotated frames in the UAVDTbenchmark. The three rows indicate the DET, MOT and SOT task, respectively. The shooting conditions of UAVs are presented in the lower right corner. The pink areas are ignored regions in the dataset. Different bounding box colors denote different classes of vehicles. For clarity, we only display some attributes.

fog. Specifically, videos shot in daylight introduce interference of shadows. Night scene, bearing dim street lamp light, offers scarcely any texture information. In the meantime, frames captured at *fog* lack sharp details so that contours of objects vanish in the background.

- **Flying Altitude** is the flying height of UAVs, affecting the scale variation of objects. Three levels are annotated, *i.e.*, *low-alt*, *medium-alt* and *high-alt*. When shooting in low-altitude ($10m \sim 30m$), more details of objects are captured. Meanwhile the object may occupy larger area, *e.g.*, 22.6% pixels of a frame in an extreme situation. When videos are collected in medium-altitude ($30m \sim 70m$), more view angles are presented. While in much higher altitude ($> 70m$), plentiful vehicles are of less clarity. For example, most tiny objects just contain 0.005% pixels of a frame, yet object numbers can be more than a hundred.
- **Camera View** consists of 3 object views. Specifically, *front-view*, *side-view* and *bird-view* mean the camera shooting along with the road, on the side, and on the top of objects, respectively. Note that the first two views may coexist in one sequence.

To evaluate DET algorithms thoroughly, we also label another 3 attributes including *vehicle category*, *vehicle occlusion* and *out-of-view*. vehicle category consists of *car*, *truck* and *bus*. vehicle occlusion is the fraction of bounding box occlusion, *i.e.*, *no-occ* (0%), *small-occ* (1% $\sim$ 30%), *medium-occ* (30% $\sim$ 70%) and *large-occ* (70% $\sim$ 100%). Out-of-view indicates the degree of vehicle parts outside frame, divided into *no-out* (0%), *small-out* (1% $\sim$ 30%) and *medium-out* (30% $\sim$ 50%). The objects are discarded when the out-of-view ratio is larger than 50%. The distribution of the above attributes is shown in Figure 2. Within an image, objects are defined as "occluded" by other objects or the obstacles in

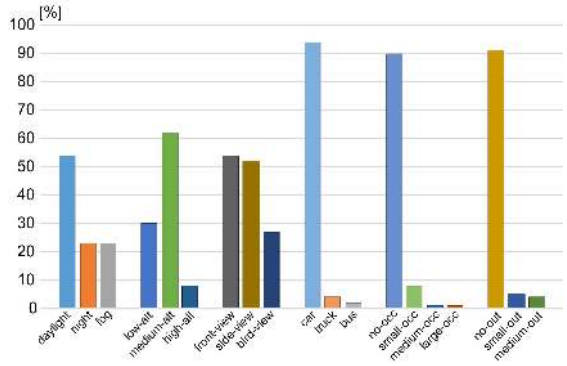

Fig. 2. The distribution of attributes of both DET and MOT tasks in UAVDT.

the scenes, *e.g.*, under the bridge; while objects are regarded as "out-of-view" when they are out of the image or in the ignored regions.

For SOT task, 8 attributes are annotated for each sequence, *i.e.*, Background Clutter (**BC**), Camera Rotation (**CR**), Object Rotation (**OR**), Small Object (**SO**), Illumination Variation (**IV**), Object Blur (**OB**), Scale Variation (**SV**) and Large Occlusion (**LO**). The distribution of SOT attributes is presented in Table 2. Specifically, 74% videos contain at least 4 visual challenges, and among them 51% have 5 challenges. Meanwhile, 27% of frames contribute to long-term tracking videos. As a consequence, a candidate SOT method can be estimated in various cruel environment, most likely at the same frame, guaranteeing the objectivity and discrimination of the proposed dataset.

Table 2. Distribution of SOT attributes, showing the number of coincident attributes across all videos. The diagonal line denotes the number of sequences with only one attribute.

	BC	**CR**	**OR**	**SO**	**IV**	**OB**	**SV**	**LO**
BC	**29**	18	20	12	17	9	16	18
CR	18	**30**	21	14	17	12	18	12
OR	20	21	**32**	12	17	13	23	14
SO	12	14	12	**23**	13	13	8	6
IV	17	17	17	13	**28**	18	12	7
OB	9	12	13	13	18	**23**	11	2
SV	16	18	23	8	12	11	**29**	14
LO	18	12	14	6	7	2	14	**20**

Notably, our benchmark is divided into training and testing sets, with 30 and 70 sequences, respectively. The testing set consists of 20 sequences for both DET and MOT tasks, and 50 for SOT task. Besides, training videos are taken at different locations from the testing videos, but share similar scenes and attributes. This setting reduces the overfitting probability to particular scenario.

2.2 Comparison with Existing UAV Datasets

Although new challenges are brought to computer vision by UAVs, limited datasets [31, 39, 22] have been published to accelerate the improvement and evaluation of various vision tasks. By exploring the flexibility of UAVs flare maneuver in both altitude and plane domain, Matthias *et al.* [31] propose a low-altitude UAV tracking dataset to evaluate ability of SOT methods of tackling with relatively fierce camera movement, scale change and illumination variation, yet it still lacks varieties in weather conditions and camera motions, and its scenes are much less clustered than real circumstances. In [39], several video fragments are collected to analyze the behaviors of pedestrians in top-view scenes of campus with fixed UAV cameras for the MOT task. Although ideal visual angles benefit trackers to obtain stable trajectories by narrowing down challenges they have to meet, it also risks diversity when evaluating MOT methods. Hsieh *et al.* [22] present a dataset aiming at counting vehicles in parking lots. However, our dataset captures videos in unconstrained areas, resulting in more generalization.

The detailed comparisons of the proposed dataset with other works are summarized in Table 1. Although our dataset is not the largest one compared to existing datasets, it can represent the characteristics of UAV videos more effectively:

- Our dataset provides a higher object density 10.52[8], compared to related works (*e.g.*, UAV123 [31] 1.00, Campus [39] 0.02, DETRAC [46] 8.64 and KITTI [19] 5.35). CARPK [22] is an image based dataset to detect parking vehicles, which is not suitable for visual tracking.
- Compared to related works [31, 39, 22] just focusing on specified scene, our dataset is collected from various scenarios in different weather conditions, flying altitudes, and camera views, *etc.*

3 Evaluation and Analysis

We run a representative set of state-of-the-art algorithms for each task. Codes for these methods are either available online or from the authors. All the algorithms are trained on the training set and evaluated on the testing set. Interestingly, some high ranking algorithms in other datasets may fail in complex scenarios.

3.1 Object Detection

The current top deep based object detection frameworks is divided into two main categories: region-based (*e.g.*, Faster-RCNN [37] and R-FCN [8]) and region-free (*e.g.*, SSD [27] and RON [25]). Therefore, we evaluate the above mentioned 4 detectors in the UAVDTdataset.

[8] The object density indicates the mean number of objects in each frame.

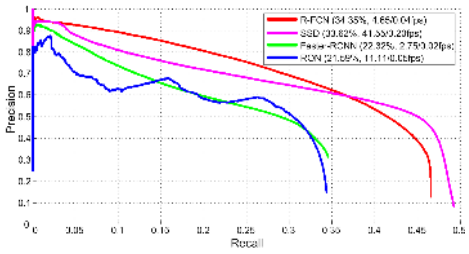

Fig. 3. Precision-Recall plot on the testing set of the UAVDT-DET dataset. The legend presents the AP score and the GPU/CPU speed of each DET method respectively.

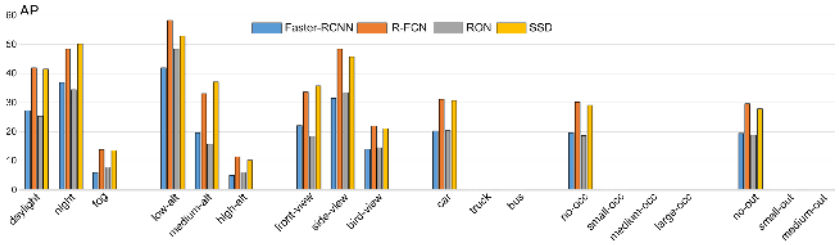

Fig. 4. Quantitative comparison results of DET methods in each attribute.

Metrics. We follow the strategy in the PASCAL VOC challenge [16] to compute the Average Precision (AP) score in the Precision-Recall plot to rank the performance of DET methods. As performed in KITTI-D [19], the hit/miss threshold of the overlap between a pair of detected and groundtruth bounding boxes is set to 0.7.

Implementation Details. We train all DET methods on a machine with CPU i9 7900x and 64G memory, as well as a Nvidia GTX 1080 Ti GPU. Faster-RCNN and R-FCN are fine-tuned on the VGG-16 network and Resnet-50, respectively. We use 0.001 as the learning rate for the first $60k$ iterations and 0.0001 for the next $20k$ iterations. For region-free methods, the batch size is 5 for 512×512 model according to the GPU capacity. For SSD, we use 0.005 as the learning rate for $120k$ iterations. For RON, we use the 0.001 as the learning rate for the first $90k$ iterations, then we decay it to 0.0001 and continue training for the next $30k$ iterations. For all the algorithms, we use a momentum of 0.9 and a weight decay of 0.0005.

Overall Evaluation Figure 3 shows the quantitative comparisons of DET methods, which shows no promising accuracy. For example, R-FCN obtains 70.06% AP score even in the hard set of KITTI-D[9], but only 34.35% in our dataset. This maybe our dataset contains a large number of small objects due

[9] The detection result is copied from http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d.

to the shooting perspective, which is a difficult challenge in object detection. Another reason is that higher altitude brings more cluttered background.

To tackle with this problem, SSD combines multi-scale feature maps to handle objects of various sizes. Yet their feature maps are usually extracted from former layers, which lacks enough semantic meanings for small objects. Improved from SSD, RON fuses more semantic information from latter layers using a reverse connection, and performs well on other datasets such as PASCAL VOC [16]. Nevertheless, RON is inferior to SSD on our dataset. It maybe because the later layers are so abstract that represent the appearance of small objects not so effectively due to the low resolution. Thus the reverse connection fusing the latter layers may interfere with features in former layers, resulting in inferior performance. On the other hand, region-based methods offer more accurate initial locations for robust results by generating region proposals from region proposal networks. It is worth mentioning that R-FCN achieves the best result by making the unshared per-ROI computation of Faster-RCNN to be sharable [25].

Attribute-based Evaluation To further explore the effectiveness of DET methods on different situations, we also evaluate them on different attributes in Figure 4. For the first 3 attributes, DET methods perform better on the sequences where objects have more details *e.g.*, *low-alt* and *side-view*. While the object number is bigger and the background is more cluttered in *daylight* than *night*, leading to worse performance in *daylight*. For the remaining attributes, the performance drops very dramatically when detecting large vehicles, as well as handling with occlusion and out-of-view. The results can be attributed to two factors. Firstly, very limited training samples of large vehicles make it hard to train the detector to recognize them. As shown in Figure 2, the number of *truck* and *bus* is only less than 10% of the whole dataset. Besides, it is even harder to detect small objects with other interference. Much work need to be done for small object detection under occlusions or out-of-view.

Run-time Performance. Although region based methods obtain relative good performance, their running speeds (*i.e.*, $<$ 5fps) are too slow for practical applications especially with constrained computing resources. On the contrary, region free methods save the time of region proposal generation, and proceed at almost realtime speed.

3.2 Multiple Object Tracking

MOT methods are generally grouped into online or batch based. Therefore, we evaluate 8 recent algorithms including online methods (CMOT [2], MDP [50], SORT [6] and DSORT [48]) and batch based methods (GOG [35], CEM [30], SMOT [13] and IOUT [7]).

Metrics. We use multiple metrics to evaluate the MOT performance. These include identification precision (IDP) [38], identification recall (IDR), and the corresponding F1 score IDF1 (the ratio of correctly identified detections over the average number of ground-truth and computed detections.), Multiple Object

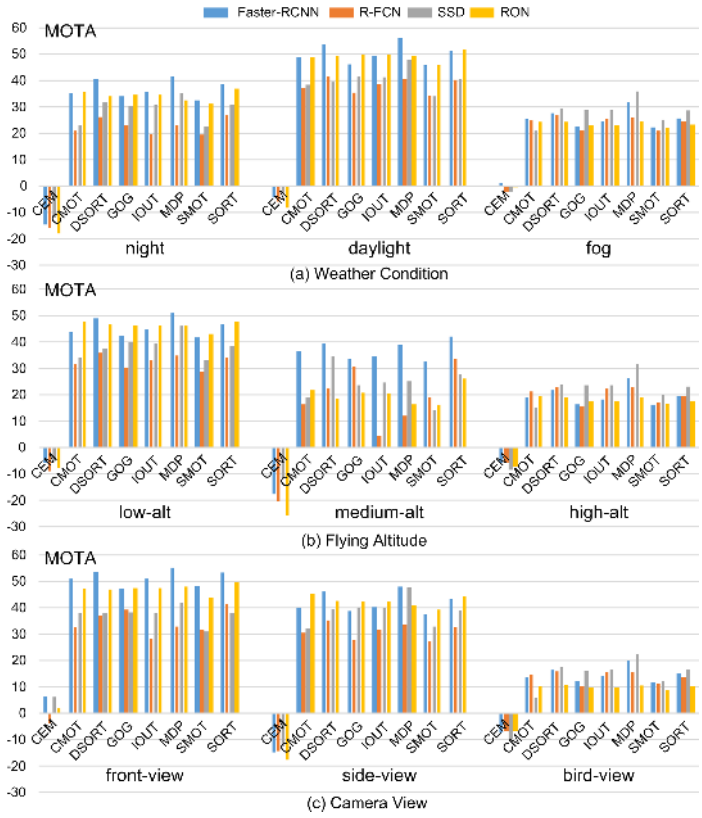

Fig. 5. Quantitative comparison results of MOT methods in each attribute.

Tracking Accuracy (MOTA) [4], Multiple Object Tracking Precision (MOTP) [4], Mostly Track targets (MT, percentage of groundtruth trajectories that are covered by a track hypothesis for at least 80%), Mostly Lost targets (ML, percentage of groundtruth objects whose trajectories are covered by the tracking output less than 20%), the total number of False Positives (FP), the total number of False Negatives (FN), the total number of ID Switches (IDS), and the total number of times a trajectory is Fragmented (FM).

Implementation Details. Since the above MOT algorithms are based on tracking-by-detection framework, all the 4 detection inputs are provided for MOT task. We run them on test set of the UAVDTdataset on the machine with CPU i7 6700 and 32G memory, as well as a NVIDIA Titan X GPU.

Overall Evaluation As shown in Table 3, MDP with Faster-RCNN has the best 43.0 MOTA score and 61.5 IDF score among all the combinations. Besides, the MOTA score of SORT in our dataset is much lower than other datasets with Faster-RCNN, *e.g.*, 59.8 ± 10.3 in MOT16 [29]. As object density is large

Table 3. Quantitative comparison results of MOT methods in the testing set of the UAVDTdataset. The last column shows the GPU/CPU speed. The best performer and realtime methods (> 30fps) are highlighted in bold font. "−" indicates the data is not available.

MOT methods	IDF	IDP	IDR	MOTA	MOTP	MT[%]	ML[%]	FP	FN	IDS	FM	Speed [fps]
Detection Input: Faster-RCNN [37]												
CEM [30]	10.2	19.4	7.0	−7.3	69.6	7.3	68.6	72,378	290,962	2,488	**4,248**	−/14.55
CMOT [2]	52.0	63.9	43.8	36.4	**74.5**	36.5	26.1	53,920	160,963	1,777	5,709	−/2.83
DSORT [48]	58.2	72.2	48.8	40.7	73.2	41.7	23.7	44,868	155,290	2,061	6,432	15.01/2.98
GOG [35]	0.4	0.5	0.3	34.4	72.2	35.5	25.3	41,126	168,194	14,301	12,516	−/436.52
IOUT [7]	23.7	30.3	19.5	36.6	72.1	37.4	25.0	42,245	163,881	9,938	10,463	−/**1438.34**
MDP [50]	**61.5**	**74.5**	**52.3**	**43.0**	73.5	**45.3**	**22.7**	46,151	**147,735**	**541**	4,299	−/0.68
SMOT [13]	45.0	55.7	37.8	33.9	72.2	36.7	25.7	57,112	166,528	1,752	9,577	−/115.27
SORT [6]	43.7	58.9	34.8	39.0	74.3	33.9	28.0	**33,037**	172,628	2,350	5,787	−/245.79
Detection Input: R-FCN [8]												
CEM [30]	10.3	18.4	7.2	−9.6	70.4	6.0	67.8	81,617	289,683	2,201	3,789	−/9.82
CMOT [2]	50.8	59.4	44.3	27.1	**78.5**	35.9	27.9	80,592	167,043	919	2,788	−/2.65
DSORT [48]	55.5	**67.3**	47.2	**30.9**	77.0	36.6	27.4	66,839	168,409	424	4,746	9.22/1.95
GOG [35]	0.3	0.4	0.3	28.5	77.1	34.4	28.6	60,511	176,256	6,935	6,823	−/433.94
IOUT [7]	44.0	47.5	40.9	26.9	75.9	**44.3**	**22.9**	98,789	**145,617**	4,903	6,129	−/**863.53**
MDP [50]	**55.8**	63.9	**49.5**	28.9	76.7	40.9	25.9	82,540	159,452	**411**	**2,705**	−/0.67
SMOT [13]	44.0	53.5	37.3	24.5	77.2	33.7	29.2	76,544	179,609	1,370	5,142	−/64.68
SORT [6]	42.6	58.7	33.5	30.2	**78.5**	29.5	31.9	**44,612**	190,999	2,248	4,378	−/209.31
Detection Input: SSD [27]												
CEM [30]	10.1	21.1	6.6	−6.8	70.4	6.6	74.4	64,373	298,090	1,530	**2,835**	−/11.62
CMOT [2]	49.4	53.4	46.0	27.2	75.1	38.3	23.5	98,915	146,418	2,920	6,914	−/0.90
DSORT [48]	51.4	**65.7**	42.2	33.6	**76.7**	27.9	26.9	**51,549**	173,639	**1,143**	8,655	15.00/3.46
GOG [35]	0.3	0.4	0.3	33.6	76.4	36.0	22.4	70,080	148,369	7,964	10,023	−/239.60
IOUT [7]	29.4	34.5	25.6	33.5	76.6	34.3	23.4	65,549	154,042	6,993	8,793	−/**976.47**
MDP [50]	**58.8**	63.2	**55.0**	**39.8**	76.5	**47.3**	**19.5**	79,760	**124,206**	1,310	4,539	−/0.13
SMOT [13]	41.9	45.9	38.6	27.2	76.5	34.9	22.9	95,737	149,777	2,738	9,605	−/11.59
SORT [6]	37.1	45.8	31.1	33.2	**76.7**	27.3	25.4	57,440	166,493	3,918	7,898	−/153.70
Detection Input: RON [25]												
CEM [30]	10.1	18.8	6.9	−9.7	68.8	6.9	72.6	78,265	293,576	2,086	**3,526**	−/9.98
CMOT [2]	57.5	65.7	51.1	36.9	**74.7**	**46.5**	**24.6**	69,109	**144,760**	1,111	3,656	−/0.94
DSORT [48]	58.3	67.9	51.2	35.8	71.5	43.4	25.7	67,090	151,007	698	4,311	17.45/4.02
GOG [35]	0.3	0.3	0.2	35.7	72.0	43.9	26.2	62,929	153,336	3,104	5,130	−/287.97
IOUT [7]	50.1	59.1	43.4	35.6	72.0	43.9	26.2	63,086	153,348	2,991	5,103	−/**1383.33**
MDP [50]	**59.9**	**69.0**	**52.9**	35.3	71.7	45.0	25.5	70,186	149,980	**414**	3,640	−/0.12
SMOT [13]	52.6	60.8	46.3	32.8	72.0	43.4	27.1	73,226	154,696	1,157	4,643	−/29.37
SORT [6]	54.6	66.9	46.1	**37.2**	72.2	40.8	28.0	**53,435**	159,347	1,369	3,661	−/230.55

in UAV videos, the FP and FN values on our dataset are also much larger than other datasets for the same algorithm. Meanwhile, IDS and FM appear more frequently. It means the proposed dataset is more challenging than existing ones.

Moreover, the algorithms using only position information (*e.g.*, IOUT, SORT) could keep fewer tracklets combining with higher IDS and FM because of absence of appearance information. GOG has the worst IDF even though the MOTA is well because of the too much IDS and FM. DSORT performs well on IDS among these methods, which means deep feature has an advantage in the aspect of representing appearance of the same target. MDP mostly has the best IDS and FM value because of their individual-wised tracker model. So the trajectories are more complete than others with the higher IDF. Meanwhile, FP values will increase by associating more objects in complex scenes.

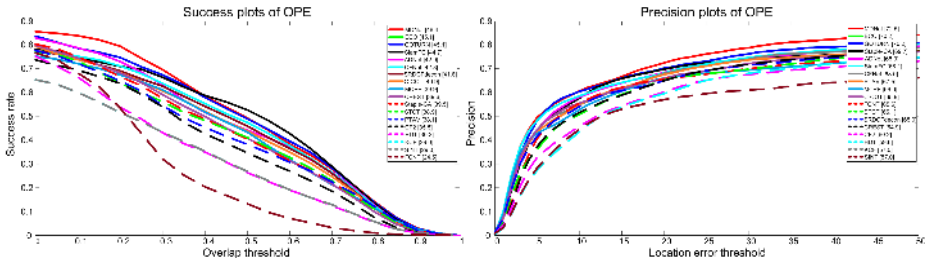

Fig. 6. The precision and success plots on the UAVDT-SOT benchmark using One-pass Evaluation [49].

Attribute-based Evaluation Figure 5 shows the performances of MOT methods on different attributes. Most methods perform better in *daylight* than *night* or *fog* (see Figure 5(a)). It is fair and reasonable that objects in *daylight* provide clearer appearance clues for tracking. In other illumination conditions, object appearance is confusing so the algorithms considering more motion clues achieve better performance, *e.g.*, SORT, SMOT and GOG. Notably, on the sequences with *night*, the performances of methods are much worse even the provided detections in *night* own a good AP score. This is because objects are hard to track with confusing environment in *night*. In Figure 5(b), the performance of most MOT methods increases with the decline of height. When UAVs capture videos in a lower height, fewer objects are captured in that view to facilitate object association. In terms of Camera Views as shown in Figure 5(c), vehicles in *front-view* and *side-view* offer more details to distinguish different targets compared with *bird-view*, leading to better accuracy.

Besides, different detection input can guide MOT methods to focus on different scenes. Specifically, the performance with Faster-RCNN is better on sequences where object details are clearer (*e.g.*, *daylight*, *low-alt* and *side-view*); while R-FCN detection offers more stable inputs for each method when sequences have other challenging attributes, such as *fog* and *high-alt*. SSD and RON offer more accurate detection candidates for tracking such that the performances of MOT methods with these detections are balanced in each attribute.

Run-time Performance. Given different detection inputs, the speed of each method varies with the number of object detection candidates. However, IOUT and SORT using only position information generally proceed at ultra-real-time speed, while DSORT and CMOT using appearance information proceed much slower. As the object number is huge in our dataset, the speed of the method processing each object respectively (*e.g.*, MDP) dramatically declines.

3.3 Single Object Tracking

The SOT field is dominated by correlation filter and deep learning based approaches [15]. We evaluate 18 recent such trackers on our dataset. These trackers can be generally categorized into 3 classes based on their learning strategy and

Table 4. Quantitative comparison results (*i.e.*, overlap score/precision score) of SOT methods in each attribute. The last column shows the GPU/CPU speed. The best performer and realtime methods (> 30fps) are highlighted in bold font. "−" indicates the data is not available.

SOT methods	**BC**	**CR**	**OR**	**SO**	**IV**	**OB**	**SV**	**LO**	Speed [fps]
MDNet [33]	**39.7/63.6**	**43.0/69.6**	**42.7/66.8**	44.4/78.4	**48.5**/76.4	**47.0**/72.4	**46.2/68.5**	**38.1/54.7**	0.89/0.28
ECO [9]	38.9/61.1	42.2/64.4	39.5/62.7	**46.1**/79.1	47.3/76.9	43.7/71.0	43.1/63.2	36.0/50.8	16.95/3.90
GOTURN [20]	38.9/61.1	42.2/64.4	39.5/62.7	**46.1**/79.1	47.3/76.9	43.7/71.0	43.7/63.2	36.0/50.8	**65.29**/11.70
SiamFC [5]	38.6/57.8	40.9/61.6	38.4/60.0	43.9/73.2	47.4/74.2	45.3/**73.8**	42.4/60.4	35.9/47.9	38.20/5.50
ADNet [52]	37.0/60.4	39.9/64.8	36.8/60.1	43.2/77.9	45.8/73.7	42.8/68.9	40.9/61.2	35.8/49.2	5.78/2.42
CFNet [43]	36.0/56.7	39.7/64.3	36.9/59.9	43.5/77.5	45.1/72.7	43.5/71.7	40.9/61.1	33.3/44.7	8.94/6.45
SRDCF [10]	35.3/58.2	39.0/64.2	36.5/60.0	42.2/76.4	45.1/74.7	41.7/70.6	40.2/59.6	32.7/46.0	−/14.25
SRDCFdecon [11]	36.0/57.4	39.0/61.0	36.6/57.8	43.1/73.8	45.5/72.3	42.9/69.5	38.0/54.9	31.5/42.5	−/7.26
C-COT [12]	34.0/55.7	39.0/62.3	34.1/56.1	44.2/79.2	41.6/72.0	37.2/66.2	37.9/55.9	33.5/46.0	0.87/0.79
MCPF [53]	31.0/51.2	36.3/59.2	33.0/55.3	39.7/74.5	42.2/73.1	42.0/73.0	35.9/55.1	30.1/42.5	1.84/0.89
CREST [41]	33.6/56.2	38.7/62.1	35.4/55.8	38.3/74.2	40.5/69.0	37.7/65.6	36.5/56.7	35.1/49.7	2.83/0.36
Staple-CA [32]	32.9/59.2	35.2/65.8	34.6/62.0	38.0/**79.6**	43.1/**77.2**	40.6/71.3	36.7/62.3	32.5/49.6	−/**42.53**
STCT [45]	33.3/56.0	36.0/61.3	34.3/57.5	38.3/71.0	40.8/69.9	37.0/63.3	37.3/59.9	31.7/46.6	1.76/0.09
PTAV [17]	31.2/57.2	35.2/63.9	30.9/56.4	38.0/79.1	38.1/69.6	36.7/66.2	33.3/56.5	32.9/50.3	12.77/0.10
CF2 [28]	29.2/48.6	34.1/56.9	29.7/48.2	35.6/69.5	38.7/67.9	35.8/65.1	29.0/45.3	28.3/38.1	8.07/1.99
HDT [36]	25.1/50.1	27.3/56.2	24.8/48.7	29.8/72.6	31.3/68.6	30.3/65.4	25.0/45.2	25.4/37.6	5.25/1.72
KCF [21]	23.5/45.8	26.7/53.4	24.4/45.4	25.1/58.1	31.1/65.7	29.7/65.2	25.4/49.0	22.8/34.4	−/39.26
SINT [42]	38.9/45.8	26.7/53.4	24.4/45.4	25.1/58.1	31.1/65.7	29.7/65.2	25.4/49.0	22.8/34.4	37.60/−
FCNT [44]	20.6/54.8	21.8/60.2	23.6/54.9	21.9/71.9	25.5/72.1	24.2/70.5	24.6/57.5	22.3/47.2	3.09/−

utilized features: I) correlation filter (CF) trackers with hand crafted features (KCF [21], Staple-CA [32], and SRDCFdecon [11]); II) CF trackers with deep features (ECO [9], C-COT [12], HDT [36], CF2 [28], CFNet [43], and PTAV [17]); III) Deep trackers (MDNet [33], SiamFC [5], FCNT [44], SINT [42], MCPF [53], GOTURN [20], ADNet [52], CREST [41], and STCT [45]).

Metrics. Following the popular visual tracking benchmark [49], we adopt the success plot and precision plot to evaluate the tracking performance. The success plot shows the percentage of bounding boxes whose intersection over union with their corresponding groundtruth bounding boxes are larger than a given threshold. The trackers in success plot are ranked according to their *success score*, which is defined as the area under the curve (AUC). The precision plot presents the percentage of bounding boxes whose center points are within a given distance ($0 \sim 50$ pixels) to the ground truth. Trackers in precision plot are ranked according to their *precision score*, which is the percentage of bounding boxes within a distance threshold of 20 pixels.

Implementation Details. All the trackers are run on the machine with CPU i7 4790k and 16G memory, as well as a NVIDIA Titan X GPU.

Overall Evaluation The performance for each tracker is reported in Figure 6. The figure shows that: I) All the evaluated trackers perform not well on our dataset. Specifically, the state-of-the-art methods such as MDNet only achieves 46.4 success score and 72.5 precision score. Compared to the best results (*i.e.*, 69.4 success score and 92.8 precision score) on OTB100 [49], a significantly large performance gap is formulated. Such performance gap is also observed when compared to the results on UAV-123. For example, KCF achieves a success score of 33.1 on UAV-123 but only 29.0 on our dataset. These results indicate that our

dataset poses new challenges for the visual tracking community and more efforts can be devoted to the real-world UAV tracking task. II) Generally, deep trackers achieves more accurate results than CF trackers with deep features, and then CF trackers with hand-crafted features. Among the top 10 trackers, there are 6 deep trackers (MDNet, GOTURN, SianFC, ADNet, MCFP and CREST), 3 CF trackers with deep features (ECO, CFNet, and C-COT), and one CF tracker with hand-crafted features namely SRDCFdecon.

Attribute-based Evaluation As presented in Table 4, the deep tracker MDNet achieves best results on 7 out of 8 tracking attributes, which can be attributed to its multiple domain training and hard sample mining. CF trackers with deep features such as CF2 and HDT fall behind due to no scale adaptation. SINT [42] does not update its models during tracking, which results in a limited performance. Staple-CA performs well on the **SO** and **IV** attributes, as its improved model update strategy can reduce over-fitting to recent samples. Most of the evaluated methods act poorly on the **BC** and **LO** attributes, which may be caused by the decline of discriminative ability of appearance features extracted from cluttered or low resolution image regions.
Run-time Performance. From the last column of Table 4, We note that I) The top 10 accurate trackers run far from real time even on a high-end CPU. For example, the fastest tracker among top 10 accurate only runs at 11.7fps and the most accurate MDNet runs at 0.28 fps. On the other hand, the realtime trackers on CPU (*e.g.*, Staple-CA and KCF), achieve success scores 39.5 and 29.0, which are intolerant for practical applications. II) When a high-end GPU card is used, only 3 out of 18 trackers (GOTURN, SiamFC, SINT) can perform in real-time. But again their best success score is just 45.1, which is not accurate enough for real applications. Overall, more work need to be done to develop a faster and more precise tracker.

4 Discussion

Our benchmark, delivering from real-life demand, vividly samples real circumstances. Since algorithms generally perform poorly on it comparing with their plausible performances with other datasets, we think this benchmark dataset can reveal some promising research trends and benefit the community. Based on the above analysis, there are several research directions worth exploring:
Realtime issues. Running speed is a crucial measurement in practical applications. Although the performance of deep learning methods surpass other methods by a large margin (especially in SOT task), the requirements of computational resources are very harsh in embedded UAV platforms. To achieve high efficiency, some recent methods [54, 47] develop an approximate network by pruning, compressing, or low-bit representing. We expect the future works count more real-time constraints not just accuracy.
Scene priors. Different methods perform the best in different scenarios. When considering scene priors in detection and tracking approaches, more robust per-

formance is expected. For example, MDNet [33] trains a specific object-background classifier for each sequence to handle varies scenarios, which make it rank the first in most datasets. We think along with our dataset this magnificent design may inspired more methods to deal with mutable scenes.

Motion clues. Since the appearance information is not always reliable, tracking methods would gain more robustness when considering motion clues. Many recently proposed algorithms make their efforts in this trend with the help of LSTM [51, 24], but still have not met with expectations. Considering with the fierce motions of both object and background, our benchmark may fruit this research trend in the future.

Small objects. In our dataset, 27.5% of objects consist of less than 400 pixels, almost 0.07% of a frame. It provides limited textures and contours for feature extraction which causes the accuracy loss of algorithms heavily based on appearance. Meanwhile, generally methods tend to save their time consuming by down-sampling images. It exacerbates the situations harshly, *e.g.*, DET methods mentioned above generally enjoy a 10% accuracy rise due to our parameters adjusting of authors provided codes and settings, mainly dealing with the size of anchors. However their performance still cannot met with expectation. We advise researchers should gain more promotions if they pay more attention on handling with small objects.

5 Conclusion

In this paper, we construct a new and challenging UAV benchmark for 3 foundational visual tasks including DET, MOT and SOT. The dataset consists of 100 videos ($80k$ frames) captured with UAV platform from complex scenarios. All frames are annotated with manually labelled bounding boxes and 3 circumstances attributes, *i.e.*, weather condition, flying altitude, and camera view. SOT dataset has additional 8 attributes, *e.g.*, background clutter, camera rotation and small object. Moreover, an extensive evaluation of most recent and state-of-the-art methods is provided. We hope the proposed benchmark will contribute to the community by establishing a unified platform for evaluation of detection and tracking methods for real scenarios. In the future, we expect to extend the current dataset to include more sequences for other high-level tasks applied in computer vision, and richer annotations and more baselines for evaluation.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China under Grant 61620106009, Grant 61332016, Grant U1636214, Grant 61650202, Grant 61772494 and Grant 61429201, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Youth Innovation Promotion Association CAS, in part by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar. Guorong Li is the corresponding author.

References

1. Mot17 challenge. https://motchallenge.net/
2. Bae, S.H., Yoon, K.: Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: CVPR. pp. 1218–1225 (2014)
3. Barekatain, M., Martí, M., Shih, H., Murray, S., Nakayama, K., Matsuo, Y., Prendinger, H.: Okutama-action: An aerial view video dataset for concurrent human action detection. In: CVPRW. pp. 2153–2160 (2017)
4. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP J. Image and Video Processing **2008** (2008)
5. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCV. pp. 850–865 (2016)
6. Bewley, A., Ge, Z., Ott, L., Ramos, F.T., Upcroft, B.: Simple online and realtime tracking. In: ICIP. pp. 3464–3468 (2016)
7. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: AVSS. pp. 1–6 (2017)
8. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: NIPS. pp. 379–387 (2016)
9. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: efficient convolution operators for tracking. CoRR **abs/1611.09224** (2016)
10. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: ICCV. pp. 4310–4318 (2015)
11. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In: CVPR. pp. 1430–1438 (2016)
12. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV. pp. 472–488 (2016)
13. Dicle, C., Camps, O.I., Sznaier, M.: The way they move: Tracking multiple targets with similar appearance. In: ICCV. pp. 2304–2311 (2013)
14. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. TPAMI **34**(4), 743–761 (2012)
15. etc., M.K.: The visual object tracking VOT2016 challenge results. In: ECCV Workshop. pp. 777–823 (2016)
16. Everingham, M., Eslami, S.M.A., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV **111**(1), 98–136 (2015)
17. Fan, H., Ling, H.: Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In: ICCV (2017)
18. Ferryman, J., Shahrokni, A.: Pets2009: Dataset and challenge. In: AVSS. pp. 1–6 (2009)
19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. pp. 3354–3361 (2012)
20. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: ECCV. pp. 749–765 (2016)
21. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. TPAMI **37**(3), 583–596 (2015)
22. Hsieh, M., Lin, Y., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. In: ICCV (2017)

23. Hwang, S., Park, J., Kim, N., Choi, Y., Kweon, I.S.: Multispectral pedestrian detection: Benchmark dataset and baseline. In: CVPR. pp. 1037–1045 (2015)
24. Kahou, S.E., Michalski, V., Memisevic, R., Pal, C.J., Vincent, P.: RATM: recurrent attentive tracking model. In: CVPRW. pp. 1613–1622 (2017)
25. Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: RON: reverse connection with objectness prior networks for object detection. In: CVPR (2017)
26. Leal-Taixé, L., Milan, A., Reid, I.D., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. CoRR **abs/1504.01942** (2015)
27. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV. pp. 21–37 (2016)
28. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: ICCV. pp. 3074–3082 (2015)
29. Milan, A., Leal-Taixé, L., Reid, I.D., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. CoRR **abs/1603.00831** (2016)
30. Milan, A., Roth, S., Schindler, K.: Continuous energy minimization for multitarget tracking. TPAMI **36**(1), 58–72 (2014)
31. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: ECCV. pp. 445–461 (2016)
32. Mueller, M., Smith, N., Ghanem, B.: Context-aware correlation filter tracking. In: CVPR (2017)
33. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR. pp. 4293–4302 (2016)
34. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV **38**(1), 15–33 (2000)
35. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. pp. 1201–1208 (2011)
36. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.: Hedged deep tracking. In: CVPR. pp. 4303–4311 (2016)
37. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
38. Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCVW. pp. 17–35 (2016)
39. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: ECCV. pp. 549–565 (2016)
40. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. TPAMI **36**(7), 1442–1468 (2014)
41. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W.H., Yang, M.: CREST: convolutional residual learning for visual tracking. CoRR **abs/1708.00225** (2017)
42. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In: CVPR. pp. 1420–1429 (2016)
43. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. CVPR (2017)
44. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: ICCV. pp. 3119–3127 (2015)
45. Wang, L., Ouyang, W., Wang, X., Lu, H.: STCT: sequentially training convolutional networks for visual tracking. In: CVPR. pp. 1373–1381 (2016)
46. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., Lim, J., Yang, M., Lyu, S.: DETRAC: A new benchmark and protocol for multi-object tracking. CoRR **abs/1511.04136** (2015)

47. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: NIPS. pp. 2074–2082 (2016)
48. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. CoRR **abs/1703.07402** (2017)
49. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. TPAMI **37**(9), 1834–1848 (2015)
50. Xiang, Y., Alahi, A., Savarese, S.: Learning to track: Online multi-object tracking by decision making. In: ICCV. pp. 4705–4713 (2015)
51. Yang, T., Chan, A.B.: Recurrent filter learning for visual tracking. In: ICCVW. pp. 2010–2019 (2017)
52. Yun, S., Choi, J., Yoo, Y., Yun, K., Choi, J.Y.: Action-decision networks for visual tracking with deep reinforcement learning. In: CVPR (2017)
53. Zhang, T., Xu, C., Yang, M.H.: Multi-task correlation particle filter for robust visual tracking. In: CVPR (2017)
54. Zhang, X., Zou, J., Ming, X., He, K., Sun, J.: Efficient and accurate approximations of nonlinear convolutional networks. In: CVPR. pp. 1984–1992 (2015)