Exploring the Effects of Blur and Deblurring to Visual Object Tracking

Qing Guo, *Member, IEEE*, Wei Feng, *Member, IEEE*, Ruijun Gao, Yang Liu, *Senior Member, IEEE*, and Song Wang, *Senior Member, IEEE*

***Abstract*— The existence of motion blur can inevitably influence the performance of visual object tracking. However, in contrast to the rapid development of visual trackers, the quantitative effects of increasing levels of motion blur on the performance of visual trackers still remain unstudied. Meanwhile, although image-deblurring can produce visually sharp videos for pleasant visual perception, it is also unknown whether visual object tracking can benefit from image deblurring or not. In this paper, we present a Blurred Video Tracking (BVT) benchmark to address these two problems, which contains a large variety of videos with different levels of motion blurs, as well as ground-truth tracking results. To explore the effects of blur and deblurring to visual object tracking, we extensively evaluate 25 trackers on the proposed BVT benchmark and obtain several new interesting findings. Specifically, we find that light motion blur may improve the accuracy of many trackers, but heavy blur usually hurts the tracking performance. We also observe that image deblurring is helpful to improve tracking accuracy on heavily-blurred videos but hurts the performance of lightly-blurred videos. According to these observations, we then propose a new general GAN-based scheme to improve a tracker's robustness to motion blur. In this scheme, a fine-tuned discriminator can effectively serve as an adaptive blur assessor to enable selective frames deblurring during the tracking process. We use this scheme to successfully improve the accuracy of 6 state-of-the-art trackers on motion-blurred videos.**

***Index Terms*— Visual object tracking, motion blur, blurred video tracking, deblurring.**

Manuscript received May 14, 2020; revised October 25, 2020; accepted November 28, 2020. Date of publication January 8, 2021; date of current version January 18, 2021. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62072334, Grant 61671325, Grant U1803264, Grant 61672376, and Grant 61906135. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weiyao Lin. *(Corresponding author: Wei Feng.)*

Qing Guo is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Beijing 100061, China, also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798.

Wei Feng and Ruijun Gao are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Beijing 100061, China, and also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China (e-mail: wfeng@ieee.org).

Yang Liu is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, and also with the Institute of Computing Innovation, Zhejiang University, Hangzhou 310027, China.

Song Wang is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Beijing 100061, China, also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China, and also with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA.

Digital Object Identifier 10.1109/TIP.2020.3045630

I. Introduction

MOTION blur caused by camera shake and object movement not only reduces the visual perception quality but also may severely degrade the performance of video analysis tasks, e.g., object tracking [1], [2]. In recent years, numerous tracking benchmarks have been proposed to evaluate how well existing trackers can handle motion blur by comparing their accuracy on videos having blurred frames [2]–[5]. However, such benchmarks do not exclude the influence of other possible interference, *e.g.*, occlusion, low resolution, illumination variation *etc*., leading to an incomplete conclusion of a tracker. In addition, with the current blur-related tracking benchmark, we cannot quantitatively evaluate trackers' robustness to different levels of motion blur, thus cannot support deep explorations of the way that motion blur affects tracking performance.

As shown in Fig. 1, given a sharp video and its blurred version for the same scene, ECO [6] locates the white billiard ball accurately on the sharp video but fails to do so when motion blur happens. In contrast, Staple_CA [7], [8] tracks the ball accurately on both videos. This situation cannot be thoroughly evaluated on blurred videos captured under different scenes. For example, the OTB benchmark [2], [3] shows that ECO has much higher accuracy than Staple_CA on its motion blur subset, which certainly does not consider the above situation. A comprehensive benchmark that fairly measures the blur robustness of trackers is necessary and will encourage the development of blur-robust trackers.

In terms of the blur robustness, a directly related task is deblurring. In recent years, deblurring methods have achieved significant progress and could be applied to various camera-based terminal devices, *e.g.*, mobile phones, smart cameras, tablet computers, *etc*., as a pre-processing module for better visual perceptive quality. In near future, we have reason to believe that the deblurring module could be a necessary part of a camera-based device. It is meaningful to study whether and how this pro-processer affects subsequent high-level computer vision tasks, *e.g.*, visual object tracking, with which we could design a new way to employ a deployed deblurring method to improve or avoid harming pre-designed trackers.

Fig. 1. Results of ECO [6] and Staple_CA [7], [8] on sharp, blurred, or deblurred videos captured from one scene. The top two rows, together with the top-right bar figure, show that ECO locates the target accurately on sharp frames while losing it on blurred ones. In contrast, Staple_CA can capture the ball in both cases. Such a situation is not considered by existing benchmarks, *e.g.*, OTB [2], [3] in which ECO has much higher accuracy than Staple_CA. The bottom two rows show that ECO misses the target on blurred frames while locating it accurately when we selectively deblur the frames.

Actually, a naive of using deblurring methods for tracking is to first deblur frames of a video and then perform trackers. However, it is known that such a naive strategy may introduce ringing artifacts, due to the Gibbs phenomenon that hurts features of raw frames and fails the tracking easily [9]–[13]. Instead of direct deblurring, many recent blur-aware trackers add different kinds of blurs to the target template, forming an augmented template set, and then locate the target at subsequent frames by matching object candidates with all of the blurred templates [10], [12]. Although effective, such trackers have high memory and computational cost. Besides, it is still unknown how to perform effective blur augmentation.

Note that, the negative effects of deblurring to visual tracking are concluded mainly based on early deblurring algorithms, and recently numerous advanced deblurring methods have been developed via deep learning, with significantly improved performance, fewer artifact noises and much faster speed [14]–[17], [17], [18], whether the recent deblurring methods are helpful for visual object tracking still remains questionable.

In this paper, we aim to analyze the effects of motion blur and deblurring methods on existing trackers and explore an effective way of using state-of-the-art deblurring methods to achieve blur-robust tracking. To this end, we first construct a totally new benchmark consisting of 100 scenes each of which contains five videos for five levels of motion blur. Then, we evaluate 25 existing trackers on the benchmark and discuss the effects of different blur levels on tracking accuracy. Meanwhile, to study the effects of deblurring methods, we consider two situations, *i.e.*, full deblurring that addresses all frames during the tracking process and selective deblurring that handles frames selectively according to the ground truth of the object's bounding boxes. All experiment results hint that selective deblurring (*e.g.*, deblurring heavy blurred frames while preserving clean or light blurred frames) is helpful for tracking accuracy enhancement. With these observations, we propose the DeblurGAN-D-based selective deblurring framework for blur-robust tracking where the discriminator (D) of a pre-trained DeblurGAN is fine-tuned and employed as a blur assessor that decides whether an incoming frame should be deblurred or not according to its blur level. Overall, our main contributions are three-fold and summarized as follows:

- We construct a Blurred Video Tracking (BVT) benchmark with a dataset containing 500 videos for 100 scenes. Each scene consists of 5 videos with different levels of motion blurs. We set three metrics to evaluate the accuracy and blur robustness of trackers.
- We extensively evaluate 25 trackers on the BVT benchmark and observe that the light motion blur improves most of the trackers, while the heavy blur hurt their accuracy significantly. We also study the effects of two state-of-the-art deblurring methods and see that deblurring can improve the tracking accuracy on heavily-blurred videos while having negative effects on the ones with light blur.
- We propose a new GAN-based tracking scheme that adopts the fine-tuned discriminator of DeblurGAN as an adaptive blur assessor to selectively deblur frames during the tracking process and improve the accuracy of 6 state-of-the-art trackers.

II. Related Work

A. Tracking Benchmarks

In recent years, numerous tracking benchmarks have been proposed for general performance evaluation or specific issues [2]–[5], [19]–[26]. The OTB [2], [3], ALOV++ [19], VOT [20], [21], [24], TrackingNet [25], LaSOT [5] and GOT-10K [26] benchmarks provide unified platforms to compare state-of-the-art trackers. More recent ones, *e.g.*, TrackingNet, LaSOT, and GOT-10K, contain a large scale of videos and cover a wide range of classes, which make training a high performance deep learning-based trackers available. Other benchmarks focus on specific applications or problems. For example, the NfS [27] benchmark consists of 100 high frame rate videos and analyzes the influence of appearance variation to deep and correlation filter-based trackers, respectively.

Among these benchmarks, OTB-2013 [3], OTB-2015 [2], TC-128 [4], and LaSOT [5] datasets contain motion blur subsets that are used to evaluate trackers' capability of handling blur. Nevertheless, the evaluation results are incomplete, since other interference that also affects the tracking accuracy is not excluded. A better solution is to compare trackers on the videos that are captured at the same scene but have different levels of motion blur to see how the trackers perform. In this paper, we construct a dataset for motion blur evaluation by

averaging the frames in high frame rate videos with different ranges, which, as a result, generates the testing videos with the same content but different levels of motion blur. By doing this, we can more objectively score the robustness of trackers and help study the effects of motion blur.

B. Motion Blur-Aware Trackers

Numerous works have studied the relationship between motion blur and visual object tracking [10]–[13], [28]–[30]. Jin *et al.* [11] have observed that matching between blurred images helps realize effective object tracking. [13], [29], [30] study how to estimate the blur kernel accurately during object tracking. Ma *et al.* [10] and Wu *et al.* [12] propose methods to integrate visual object tracking with the motion blur problem through sparse representation and realize blur-robust trackers.

The above works are based on the observation that deblurring methods introduce adverse effects to frames and corrupt the features, which, however, have achieved significant progress in recent years. Whether the latest deblurring works are helpful for object tracking remains unknown. Recent work [28] shows that motion blur is helpful by providing additional motion information of the target. However, it does not discuss the effects of different levels of motion blur to object tracking.

C. Other State-of-the-Art Trackers

The latest tracking works focus on constructing powerful appearance models to realize high-performance tracking. We can coarsely split recent works into three categories including correlation filter (CF) based [6], [7], [31]–[35], [83], [84] classification&updating based [36]–[39] and Siamese network or matching based [40]–[50], [85] trackers. Some trackers [51], [52] further employ extra information, *e.g.*, thermal infrared, and saliency detection [53], [54], for accurate tracking. [55] proposes to tune the hyperparameters of the Siamese network and CF-based trackers, achieving significant performance improvement. Although the above trackers have achieved great performance improvement on benchmarks, there is no specific benchmark that can evaluate their ability to handle different levels of motion blur. Note that, evaluating the effects of motion blur to single-object tracking also benefits multiple-object tracking [56], [57] where the single-object trackers are usually employed for data association across frames.

D. GAN Based Methods

Generative adversarial networks (GANs) [58] are to train two competitors, *i.e.*, the discriminator, and the generator. The generator is to produce fake samples that can fool the discriminator. The discriminator is to separate fake samples from real ones. With recent studies [59], [60] to alleviate training problems of GAN [61], it has helped achieve significant progress in areas of deblurring [14], superresolution [62] and image painting [63], [64] and other related problems.

Nevertheless, most of the GAN-based methods regard the discriminator as a part of the loss function to train the generator and discard it during testing time. In this paper, we find that the discriminator trained for DeblurGAN [14] can score the blur level of motion blur and help realize selective deblurring for blur-robust tracking.

III. Blurred Video Tracking (BVT) Benchmark

In this section, we introduce how to construct the dataset of Blurred Video Tracking (BVT) benchmark. We first collect 100 high frame-rate (240 fps) videos having small temporal variation across neighboring frames and use them to generate videos with different blur levels by averaging neighboring frames to simulate the real motion blur generation. The number of averaged neighboring frames decides the blur level. Note, in recent studies of deblurring [15], [16], such blur generated by averaging neighboring frames is called 'realistic' blur in contrast to synthetic blur produced by algorithms. It has been shown that deblurring models trained on the 'realistic' blurred data do better than those trained on synthetic ones [16]. With the built dataset, we then set up three metrics to evaluate the accuracy and blur robustness of trackers.

A. Dataset

1) High Frame-Rate Video Collection: We initially collect 164 high frame-rate videos captured at 240 fps. Specifically, we shot 38 and 19 videos via GoPro and iPhone 8 Plus with a stabilizer, respectively, and borrowed 100 videos from NfS dataset [27] and seven videos from the internet. For all videos, we annotate each frame of them with an axis-aligned bounding box that indicates the object to be tracked.

Although frames in the NfS videos are usually sharp, there exist abrupt shake among neighboring frames caused by the fast-moving targets and cameras. These frames may lead to artifacts, *e.g.*, multiple exposures when we use them to simulate blurred videos by averaging adjacent frames. We then exclude these videos according to target annotations and optical flow. That is, for a high frame-rate video, we will remove it from the final dataset if it contains neighboring frames where the center distance between their annotations are larger than 2 pixels or the optical flow magnitude is smaller than a threshold. Finally, we got 100 high frame-rate videos. The interested targets include person (basketball player, walker, soccer, etc.), animal (ant, dog, duck, etc.), vehicle (bike, minibus, SUV, etc.), and generic objects (ball, bottle, mobile, etc.), and are captured in various scenes, *e.g.*, office, swimming pool, playground, and noisy street.

2) Multi-Level Blurred Video Generation: Given a 240 fps video denoted as $\mathcal{V} = \{\mathbf{I}_t\}_1^T$, we aim to produce five 15 fps[1] videos whose frames have different motion blur levels. To this end, we temporally sample sharp frames from $\mathcal{V}$ at every 16 frames and blur these sharp frames by averaging their L neighboring frames to produce a blurred 15 fps video, *i.e.*, $\tilde{\mathcal{V}}^L = \{\tilde{\mathbf{I}}_t^L\}_1^{\tilde{T}}$ where $\tilde{\mathbf{I}}_t^L = \text{avg}(\{\mathbf{I}_{t-\frac{L}{2}}, \ldots, \mathbf{I}_t, \ldots, \mathbf{I}_{t+\frac{L}{2}}\})$. The number L determines the blur level of $\tilde{\mathcal{V}}^L$, *i.e.* a

[1]We do not generate $\geq$ 30 fps videos since these videos have small temporal variation across neighboring frames while trackers can obtain high and similar accuracy on all blur levels [27]. As a result, we cannot analyse how different motion blurs affect trackers.

larger L leads to a severer blur. We consider 5 blur levels for $L \in \{0, 2, 4, 8, 16\}$, respectively, where $L = 0$ means the frames are exactly the original sharp ones in $\mathcal{V}$. Note that, we also evaluate all trackers on more blur levels with $L \in \{0, 2, 4, 6, 8, 10, 12, 14, 16\}$ and observe similar results on the subsequent experiments.

There are commonly two ways to construct blurred frames [15], [65], *i.e.*, blur-kernel-based synthetic strategy, and 'realistic'-based strategy by averaging sharp frames during shutter time. In this work, we use the second strategy due to the following reasons: *First*, the second strategy can generate realistic and spatially varying blurs since it follows the principle of camera imaging and the results are usually named 'realistic' blur [15], [16]. Its effectiveness has been demonstrated in a wide range of blur-related works [15], [16], [65], [66], [16] shows that the deblurring model trained on 'realistic' blurred frames does better than the one trained on the synthetic frames. Kim [65] uses the strategy to construct an evaluation dataset for deblurring. A more recent work [66] adopts the strategy to generate a dataset for training a motion-blur synthesization model. *Second*, the first strategy does not meet the requirements of studying the visual object tracking task: 1) it cannot model challenging cases such as occlusion, complex deformation, or depth variations that frequently happen in visual tracking. 2) kernel estimation is subtle and sensitive to noise and saturation, which may lead to artifacts. 3) calculating spatially varying kernels for each pixel in a dynamic scene requires an amount of memory and computation [15].

3) Ground Truth Generation: We have bounding box annotations on all the sharp frames in the original high frame-rate videos while each blurred frame in $\tilde{\mathcal{V}}^L$, *i.e.*, $\tilde{\mathbf{I}}_t^L$, corresponds to a sharp one, *i.e.*, $\mathbf{I}_t$, in $\mathcal{V}$. We can simply take those sharp frames' annotations as the bounding box annotations of corresponding blurred ones. A similar strategy is also adopted in building a deblurring training dataset where a sharp frame is set as the deblurring ground truth of the corresponding blurred frame that is generated by averaging the sharp frame and its neighboring frames [16].

4) Initial Frame Selection: In a general tracking process, a target template is cropped from the first frame of a video and used to initialize a target appearance model for the subsequent tracking. To track a target in one of our blurred videos, if we also use the first frame of the blurred video to initialize a tracker, the target appearance model would contain motion blur information. As a result, the tracker gets high accuracy easily, even if it is not good at handling motion blur, and trackers' robustness evaluation results would be affected. To avoid this problem, we set the first frame of the high frame rate video, *i.e.*, $\mathcal{V}$, as the initial frame of $\bar{\mathcal{V}}^L$.

Following the above setups, we obtain blurred videos that make up a new dataset denoted as $\mathcal{S}$, which contains 500 videos and consists of 5 subsets, *i.e.*, $\mathcal{S} = \{\mathcal{S}^L | L = 0, 2, 4, 8, 16\}$, corresponding to 5 different levels of motion blur. Fig. 2 shows three cases of blurred frames in 5 different levels. Clearly, through temporal averaging on high frame-rate frames, we obtain 'realistic' blurred videos in which the blur is directly related to the object and camera motion patterns.

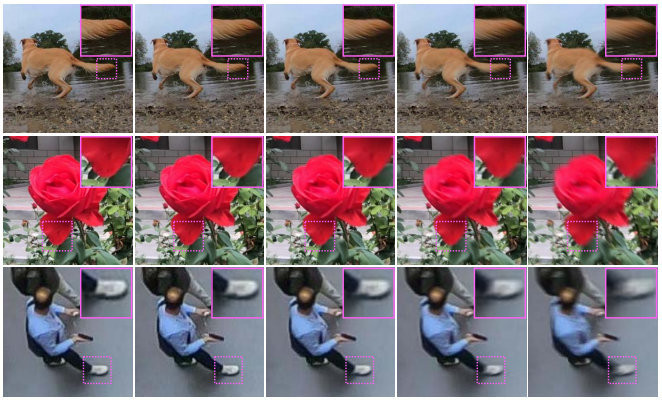

Fig. 2. Examples of frames that are blurred in 5 levels. '$L = 0$' represents the original video captured at 240 fps and it has the least serious blur.

B. Metrics

We set three metrics for the blur robustness evaluation based on the success metric defined in [2]. Precisely, we first calculate the intersection over union (IoU) between predicted and annotated bounding boxes at each frame of a subset $\mathcal{S}^L$. We then draw a success plot that presents the percentage of bounding boxes whose IoU is larger than given thresholds. The area under curve (AUC) of the success plot compares different trackers on the subset $\mathcal{S}^L$ and is denoted as A^L. Given a tracker, we can draw a blur robustness plot where the X-axis represents five subsets and Y-axis denotes the five AUC scores on the subsets. We can rank compared trackers according to the average and standard variance of AUC scores, respectively. The average of five AUC scores measures the absolute accuracy of a tracker on different blurred videos while the standard variance represents the robustness.

In addition, a blur-robust tracker should be insensitive to the blur level variation, that is, the tracking results (*e.g.*, IoUs) should not decrease under severe blur in particular on the frames that tracker can achieve high accuracy. Under this intuition, we propose a new metric named normalized robustness curve to evaluate the robustness of a tracker by testing the influence of motion blur on the sharp frames where the object can be successfully localized. To this end, we first evaluate a track on the sharp video subset $\mathcal{S}^0$ and collect frames on which the IoUs are larger than 0.5 (*i.e.*, the object is successfully localized), and we represent those frames as $\mathcal{I}_{\text{succ}}^0$. Each frame in $\mathcal{I}_{\text{succ}}^0$ has four blurred versions on other four subsets, *i.e.*, $\mathcal{S}^{\{2,4,8,16\}}$. Then, the corresponding blurred frames of $\mathcal{I}_{\text{succ}}^0$ under blur level L make up the set $\mathcal{I}_{\text{succ}}^{L\in\{2,4,8,16\}}$. We calculate the average IoUs on $\mathcal{I}_{\text{succ}}^{L\in\{0,2,4,8,16\}}$, respectively, and the results lead to a vector, *i.e.*, $[u_0, u_2, u_4, u_8, u_16]$ where u_L denotes the mean IoU of $\mathcal{I}_{\text{succ}}^L$. Finally, we finally get a normalized vector by

$$\mathbf{u} = \frac{[u_0, u_2, u_4, u_8, u_{16}]}{u_0}, \tag{1}$$

where $\mathbf{u}$ denotes a normalized robustness curve (NRC). The second subfigure in Fig. 3 shows the NRCs of evaluated trackers. Note that, the number of successfully tracked frames

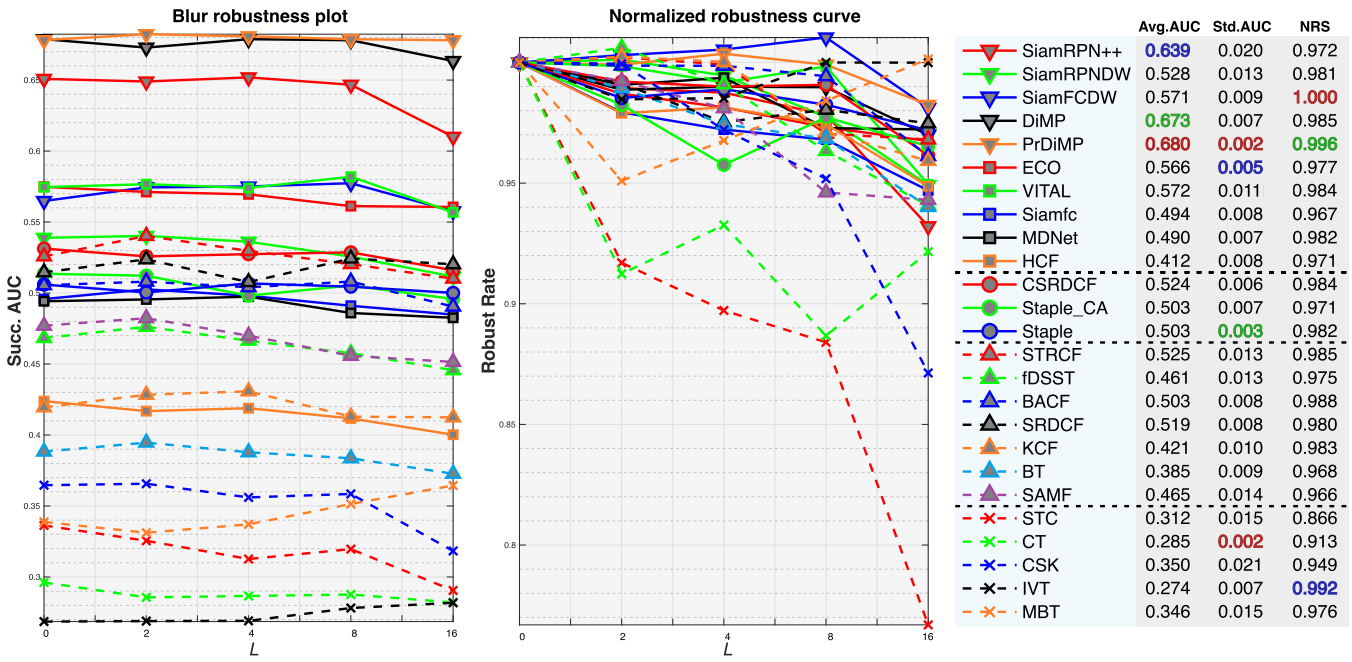

Fig. 3. Evaluation results of 25 trackers on the BVT benchmark. The left subfigure shows the blur robustness plot of each tracker. The middle subfigure presents the normalized robustness curves of all trackers. The right subfigure displays the normalized robustness score (NRS), average AUC, and its standard variation on 5 subsets of each tracker respectively. The best three results are highlighted by red, green, and blue, respectively.

on $\mathcal{I}_{\text{succ}}^{\{2,4,8,16\}}$ is less than or equal to that on $\mathcal{I}_{\text{succ}}^{0}$. Hence, u_L with $L > 0$ is usually smaller than or equal to u_0, leading to most of the NRCs in Fig. 3 are not a larger one, and NRC could be larger than one only when the tracker on $\mathcal{I}_{\text{succ}}^{\{2,4,8,16\}}$ achieves larger IoUs than on $\mathcal{I}_{\text{succ}}^{0}$, as the NRC of SiamFCDW shown in Fig. 3. The average of all elements in $\mathbf{u}$ is called the normalized robustness score (NRS). If the NRS of a tracker approximates to 1, it means that the tracker is not affected by the motion blur and can still locate the target on blurred versions of $\mathcal{I}_{\text{succ}}^{0}$.

Note that, the standard variation of AUC (Std. AUC) and NRS may lead to different robustness results since they evaluate trackers from two different perspectives. Std. AUC is a global metric and measures how different levels of motion blur influence a tracker on the whole dataset. NRS is calculated according to the results of successfully tracked sharp frames (*i.e.*, $\mathcal{I}_{\text{succ}}^{0}$) and is a local metric that evaluates if a tracker can still locate a target when motion blur is added to $\mathcal{I}_{\text{succ}}^{0}$. Intuitively, a blur-robust tracker should have very low Std. AUC with its NRS approximating to 1. For our robustness evaluation, it is better to regard Std. AUC as the main metric and use NRS when two trackers have similar Std. AUCs.

IV. Evaluation Results

With the proposed BVT benchmark, we extensively evaluate 25 trackers and analyze their blur robustness. Meanwhile, we use two state-of-the-art deep deblurring methods to handle the blurred subsets of the BVT benchmark and discuss whether and how these methods can help improve tracking performance. More specifically, we construct two deblurring strategies for visual object tracking, *i.e.*, full deblurring, and selective deblurring. The first one deblurs all incoming frames while the second one selectively processes frames according to the center localization error between predictive and ground truth bounding boxes. The experimental results motivate a novel blur-robustness tracking framework in Sec. V where the discriminator of a pre-trained DeblurGAN is employed as the blur assessor to guide the selection of deblurring.

A. Effects of Blur to Tracking

1) Trackers: We evaluate 25 trackers on the proposed benchmark and categorize them into 4 classes according to representations they used: trackers using intensity-based features,[2] *i.e.*, IVT [67], CT [70], CSK [68], STC [69] and MBT [10], trackers based on HoG features, *i.e.*, BT [71], KCF [72], SAMF [73], SRDCF [74], fDSST [75], BACF [33] and STRCF [31], trackers with deep features, *i.e.*, HCF [76], ECO [6], MDNet [37], Siamfc [40], VITAL [38], SiamRPN++ [77], SiamFCDW [78], SiamRPNDW [78], DiMP [79], and PrDiMP [80], and trackers using mixed features, *i.e.*, Staple [7], Staple_CA [8], and CSRDCF [32].

2) Overall Results: We present the evaluation results in Fig. 3 and 4. In general, the accuracy of most trackers decreases as the blur level increases. In terms of the average AUC, PrDiMP [80] achieves the highest accuracy while DiMP [66] and SiamRPN++ [77] are in the second and third place, respectively, since these trackers employ deep features as object representations and are equipped with well-designed architectures or online learning strategies. Among trackers based on hand-crafted features, CSRDCF [32], STRCF [31], and SRDCF [74] is in the first, second, and third places, respectively. Moreover, these trackers

[2]Here, the intensity-based features consist of the template used by IVT [67], CSK [68] and STC [69], and haar-like features used by CT [70].

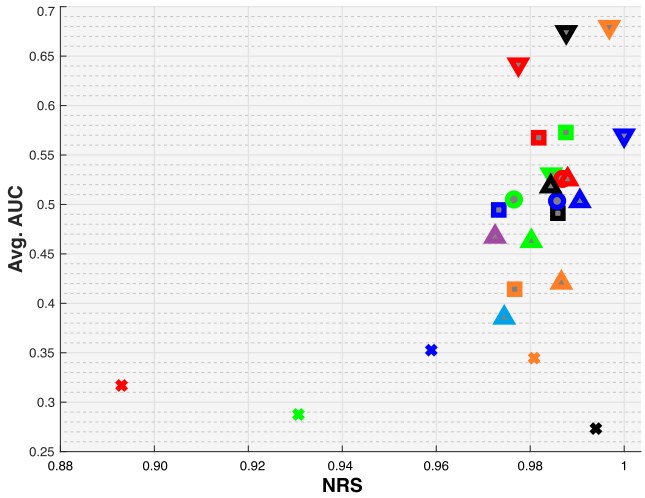

Fig. 4. Normalized robustness score (NRS) and average AUC of 25 trackers. The legend is the same as that of Fig. 3.

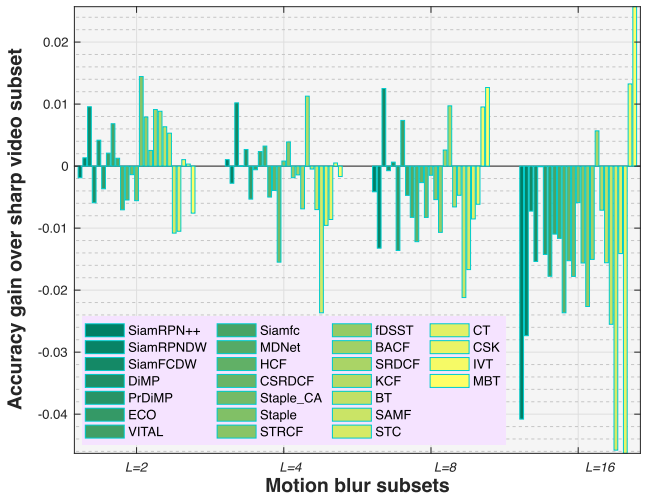

Fig. 5. AUC gains of blurred subsets, *i.e.*, $\mathcal{S}^{\{2,4,8,16\}}$, over sharp video subset, *i.e.*, $\mathcal{S}^0$ for all compared trackers.

are better than MDNet [37] and HCF [76] that use deep features. The trackers using intensity-based features have much lower accuracy than others due to the less discriminative features. However, what is surprising is that the AUCs of IVT [67] and MBT [10] gradually increase as the blur level becomes larger, which means serious motion blur improves their ability to localize targets. These results cannot be concluded from existing benchmarks and validate the effectiveness of MBT that is a specifically designed tracker for handling motion blur.

In terms of the robustness evaluation, SiamFCDW [78], PrDiMP [80] and IVT [67] are the three highest normalized robustness score (NRS). PrDiMP [80], CT [70], Staple [7], and ECO [6] gets the smallest standard variance of AUC, which means they are robust to huge blur level variation. MBT [70] has lower NRS and larger standard variance than IVT due to the improved accuracy on serious blur levels.

As shown in Fig. 4, considering both average AUC and NRS, we find that PrDiMP [80] and DiMP [66] achieve the best performance in both accuracy and blur robustness. Besides, ECO [6] and VITAL [38] also have a good balance between accuracy and blur robustness.

According to blur robustness plots, we observe that the ranks of trackers have a great difference on 5 subsets. For example, SRDCF [74] has smaller AUC scores than CSRDCF [32] and STRCF [31] on $\mathcal{S}^{\{0,2,4\}}$ while being the best one on $\mathcal{S}^{16}$. We can find similar results on ECO [6], MBT [76], MDNet [81], and Staple [7].

Note that, we also conduct above experiments on more blur levels with $L \in \{0, 2, 4, 6, 8, 10, 12, 14, 16\}$ and observe similar results. Besides, the trackers' results on moderate blur levels (*i.e.*, $L \in \{10, 12, 14\}$) is similar with the ones on heavy blur level, *i.e.*, $L = 16$.

In summary, we have the following observations: *Simply comparing trackers on a single subset is not enough to conclude their abilities to handle motion blur. The accuracy and blur robustness of trackers are dependent on the features they used. A large rate of deep feature-based trackers obtain high accuracy but are somehow sensitive to severe blur. Some intensity-based trackers,* e.g., *IVT, and MBT, can leverage severe motion blur to improve their tracking accuracy. It is necessary and interesting to explore possible combination strategies to take both advantages.*

3) Benefits of Light Motion Blur: According to blur robustness plots shown in Fig. 3, a lot of trackers obtain higher AUC on lightly-blurred subsets, *e.g.*, $\mathcal{S}^{\{2,4\}}$, than on $\mathcal{S}^0$. This infers that *the light motion blur has positive effects on tracking accuracy.* To better understand this observation, for each tracker, we calculate the AUC gain of blurred subsets, *i.e.*, $\mathcal{S}^{\{2,4,8,16\}}$, over the sharp version, *i.e.*, $\mathcal{S}^0$. As shown in Fig. 5, on the lightly-blurred subsets, *i.e.*, $\mathcal{S}^2$, and $\mathcal{S}^4$, there are 12 and 7 trackers that have positive gains. Such numbers reduce to 5 and 3 on heavily-blurred subsets, *i.e.* $\mathcal{S}^8$ and $\mathcal{S}^{16}$, respectively. Hence, light motion blur does help most of the compared trackers obtain higher accuracy. There are two possible reasons for this observation: 1) Lightly-blurred videos generated by averaging neighboring high rate frames contain more effective information for separating the target from the background. 2) Temporal average helps reduce videos' noise and improve trackers' accuracy.

To clarify which reason leads to the accuracy gain, we provide two-fold analyses and tests. First, although temporal averaging may reduce the noise of blurred frames, if exist, seems helpful to tracking performance, our BVT dataset is composed of HD (1280×720) video sequences with clean frames and very low-level noises. Thus, the noise level is not significantly reduced by temporal averaging.

Second, to further analyze whether the light blur or reduced noise helps enhance tracking accuracy, we generate multi-blurred and multi-noised videos based on high frame-rate videos from NfS dataset [27]. We do not use the self-collected videos of the BVT dataset to eliminate the influence of the dataset itself. Specifically, we add multi-level Gaussian noises (with standard deviations of 0, 10, 20, and 30) to all blurred frames and let all blur levels share the same noise degree. We take the STRCF [31] as an example[3] and run it

[3]STRCF is a representative CF-based tracker that uses HOG features and has a slight improvement on the blur level $L = 2$ in the BVT.

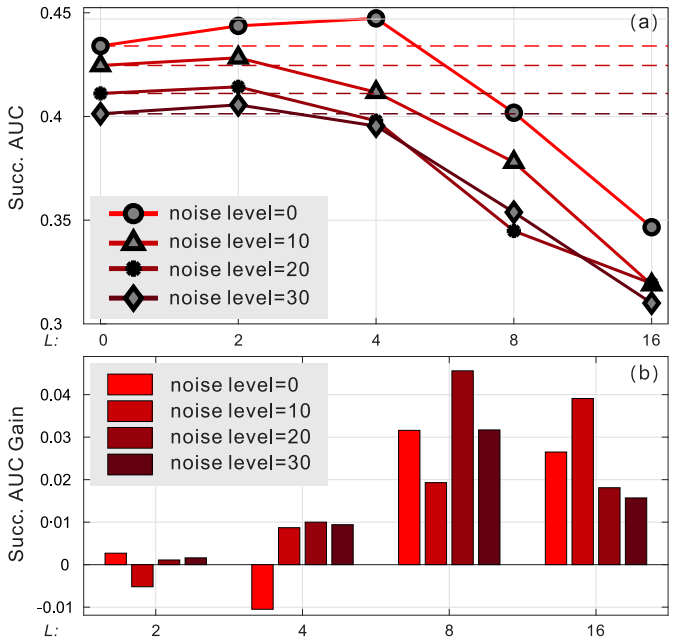

Fig. 6. Effects of noise, blur, and deblurring to visual tracking. (a) shows the Succ. AUC of STRCF on datasets with five blur levels and four noise levels. (b) shows the Succ. AUC gain of STRCF before and after full debulrring.

on the noise-blurred subsets. As shown in Fig. 6 (a), we see that STRCF gets higher accuracy on $L = 2$ than on $L = 0$ for all noise levels, verifying the statement that even with severe noises light motion blur still benefits tracking accuracy.

We also observe that ECO using deep features always obtains negative gains on all subsets where the gain is gradually enlarging as the blur becomes severer. We have similar results on HCF and CSRDCF, although they obtain high AUCs on $\mathcal{S}^0$. In contrast, IVT and MBT have the highest positive gains on severe blur subsets, *i.e.*, $\mathcal{S}^{16}$, which further demonstrate the importance of their way in handling motion blur.

In summary, we have the following observations: *Light motion blur helps most of the trackers achieve higher accuracy while heavy blur significantly reduces the performance of almost all trackers.*

B. Effects of Deblurring to Tracking

In this subsection, we study whether and how the state-of-the-art deep deblurring methods could improve the accuracy of trackers under different motion blur levels. Particularly, we consider two questions: Does deblurring all frames equally benefit to tracking accuracy enhancement on different blur levels? Does deblurring frames selectively help blur-robust tracking? To answer the first question, we simply deblur all frames via the state-of-the-art deblurring methods before feeding them to a tracker. In terms of the second question, we conduct an exploratory experiment where we selectively deblur incoming frames with the guidance of ground truth bounding boxes. Intuitively, at each frame, we estimate two bounding boxes based on the original frame and the deblurred one and use the ground truth bounding box to decide which one is more accurate and should be selected as the final output. In the following, we detail how to implement the experiments and analyze the results extensively, which inspires a novel blur-robust tracking framework in Sec. V. Note that, we use ground truth bounding boxes to clarify the possibility of employing deblurring methods for blur-robust tracking and the proposed new tracking framework in Sec. V does not rely on any extra information.

1) Setups: Early deblurring methods are slow and not suitable for real-time tracking. We select two deep deblurring methods, *i.e.*, DeblurGAN [14] and SRN, [82], that run much faster via GPU [4] and achieve state-of-the-art deblurring performance. Given a tracker, we use a deblurring method to construct two variants of the tracker. The first one is to deblur all frames before feeding them to a tracker, and we name it the *full deblurring*-based tracking method. The second one is to selectively deblur frames during the tracking process with the guidance of ground truth bounding boxes. Specifically, given an incoming frame, we deblur the frame and estimate the object's bounding boxes on the original and deblurred frames, respectively. Then, we set the result of having smaller center localization errors to the ground truth bounding box as the final output. Through the two deblurring strategies, we get four variants for each tracker and denote them as '*_gan', '*_srn' for full deblurring-based ones, and '*_ganslt', '*_srnslt' for selective deblurring-based methods, respectively, where '*' represents the name of a tracker. We test these variants on four blurred video subsets, *i.e.*, $\mathcal{S}^{\{2,4,8,16\}}$.

For a comprehensive study, we select six representative trackers including the ones that get high accuracy on our BVT benchmark, *i.e.*, STRCF [31] and ECO [6], a Siamese network-based tracker, *i.e.*, Siamfc [40], CF trackers with hand-crafted features, *i.e.*, fDSST [75] and Staple_CA [7], [8], and a motion blur-aware tracker, *i.e.*, MBT [10].

2) Cons of Full Deblurring: As shown in Fig. 7, when we deblur all frames via DeblurGAN, most of trackers' accuracy decreases on lightly-blurred subsets, *i.e.*, $\mathcal{S}^{\{2 \text{ or } 4\}}$, while increasing on the severely-blurred ones, *i.e.*, $\mathcal{S}^{\{8 \text{ or } 16\}}$. For example, the AUC of STRCF_gan is much smaller than that of STRCF on $\mathcal{S}^{2,4}$ while being larger on $\mathcal{S}^{8,16}$. We can find similar results when using the SRN method. Such observations encourage that we should perform deblurring on severely-blurred frames and pass the ones with light blur when we use DeblurGAN and SRN to improve the tracking accuracy.

For the motion blur-aware tracker, *i.e.*, MBT, we find the almost opposite conclusion, that is, MBT_gan and MBT_srn get much better results on the lightly-blurred subsets while being lightly worse on severely-blurred ones, which implies that we should deblur the lightly-blur frames while passing the heavily-blurred ones for MBT tracker.

In summary, we have the following observations: *State-of-the-art deep deblurring methods,* i.e. *DeblurGAN [14] and SRN [82], usually result in tracking accuracy decreasing on*

[4]DeblurGAN takes average 0.02 s to deblur search regions that are about 5 times larger than targets.

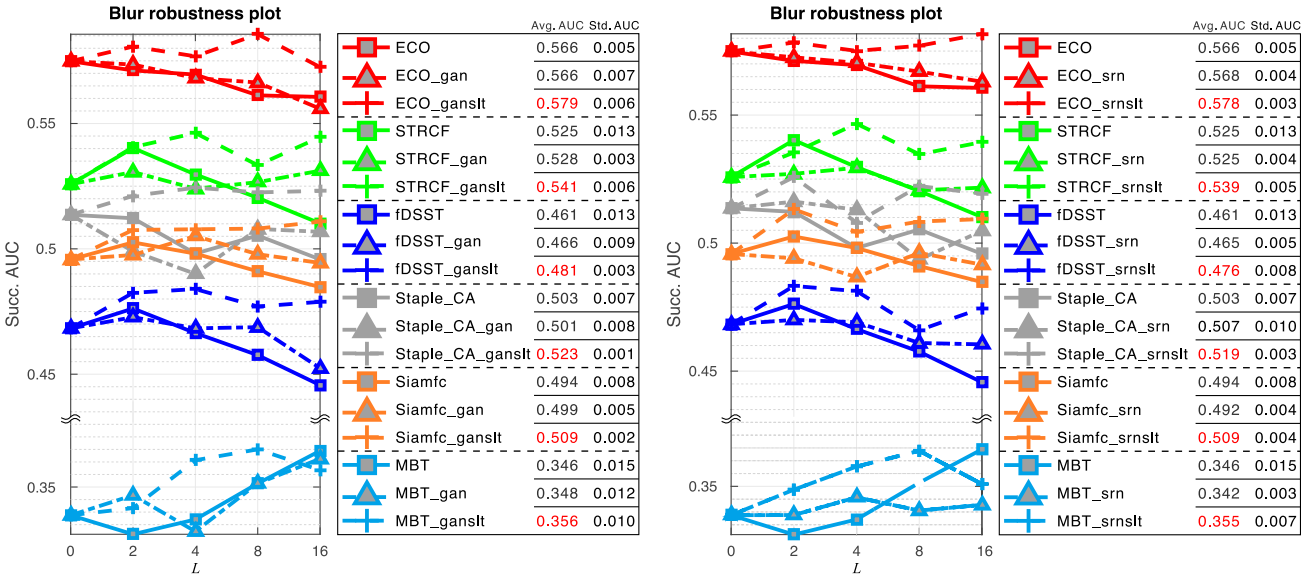

Fig. 7. Evaluation results of 6 typical trackers and their four variants. DeblurGAN [14] and Scale-recurrent network (SRN) [82] is used to cope with the blurred frames respectively. '*_gan' and '*_srn' denote trackers deblurring each frame via DeblurGAN and SRN respectively. '*_ganslt' and '*_srnslt' are methods that selectively deblur frames according to the localization error of trackers.

lightly-blurred videos while having positive effects on the ones containing heavy motion blur.

We further evaluate the influence of noise on the full deblurring by testing a representative tracker, *i.e.*, STRCF, and evaluate it on multi-blurred and multi-noised videos from NfS dataset. Specifically, each frame from the tested dataset is added Gaussian noises (with standard deviations of 0, 10, 20, and 30) and then deblurred by DeblurGAN before fed to STRCF. The Succ. AUC gains of STRCF before and after full deblurring are presented in Fig. 6 (b) and we observe that: in all noise levels, full deblurring harms the accuracy or contributes limited improvements to STRCF on lightly blurred videos while always benefiting the tracker on videos with heavy motion blur.

3) Pros of Selective Deblurring: According to observations in Section. IV-A and IV-B, selective deblurring, *i.e.*, only performing deblurring at some frames, should improve tracking accuracy. To validate this assumption, we selectively deblur an incoming frame according to localization errors during the tracking process with the supervision of annotations. Specifically, for a blurred frame t, we first use DeblurGAN or SRN to deblur it and then predict the target position according to blurred and deblurred frames, respectively. We thus obtain two bounding boxes whose center localization errors to the ground truth are calculated. The result with higher precision is saved as the final output. We name the above methods as '*_ganslt' or '*_srnslt' for DeblurGAN and SRN, respectively.

Fig. 7 shows that selective deblurring with both DeblurGAN and SRN improves AUC scores of most of the trackers significantly. Furthermore, we see that trackers with selective deblurring generally get higher gain over their original versions on heavily-blurred videos than on light ones.

In summary, we have the following observations: *Selective deblurring improves tracking performance significantly. Accuracy gains incrementally increase with the growing motion blur level and generally reaches the maximum at the most heavily-blurred video subset.*

V. Blur-Robust Tracking via DeblurGAN-D

A. DeblurGAN-D as Blur Assessor

DeblurGAN [14] uses a critic network as its discriminator to predict scores of sharp images and GAN-deblurred images. Wasserstein distance between the scores is set as the loss to train the discriminator (D) and the GAN (G). In general, the D only works at the training process and is discarded at testing time. Here, we explore how to use it as a blur assessor. From the view of training D, we tune D's parameters to distinguish between sharp images and GAN-deblurred images. During the training stage, these deblurred images have different blur levels since the deblurring ability of the G is gradually improved. As a result, the discriminator has the ability to make a distinction between sharp and blurred images.

As shown in Fig. 8, we calculate discriminator outputs of frames in four videos that have different blur levels. Clearly, the heavily-blurred video, *i.e.*, $L = 16$, has the smallest value while the sharp one, *i.e.*, $L = 2$, has the highest score. Hence, the discriminator of DeblurGAN, *i.e.*, DeblurGAN-D, is able to score the blur levels of frames and will help decide when we should do deblur during the tracking process.

B. Fine-Tuning DeblurGAN-D

Although we have shown DeblurGAN-D can score the blur degree of a frame, it easily fails to discriminate motion blur degrees when their visual difference is small. As shown in Fig. 9, DeblurGAN-D cannot rank the blur degree of frames properly. This is because DeblurGAN-D is originally designed to compare the sharp and deblurred images, which has a gap with the task of assessing blur degrees.

To alleviate the above problem, we propose to fine-tune DeblurGAN-D with blur & deblur image pairs. Specifically,

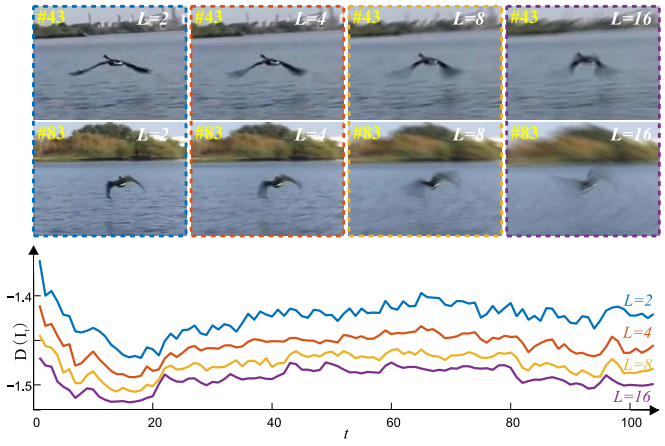

Fig. 8. Pipline Outputs of the discriminator of DeblurGAN, *i.e.* D(·), on bird sequences that contain 4 levels of motion blur.

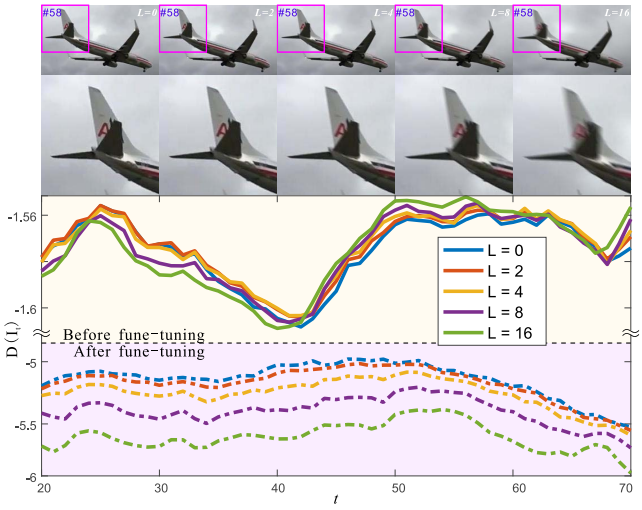

Fig. 9. Comparing the fine-tuned discriminator with the original one on airplane sequences.

we select 20 high frame-rate video clips from the NfS dataset, which are not included in our dataset. Then, we generate 80 blurred videos via the same way detailed in Section 3.1 and get 32,304 frames. Each high frame-rate video corresponds to 4 blurred videos with 4 levels of blur, respectively. We use the DeblurGAN-G to deblur these frames and get 32,304 blur & deblur image pairs. Using these pairs as training data, we particularly fine-tune the discriminator via the same adversarial loss of DeblurGAN with the fixed generator. As shown in Fig. 9, compared with the original discriminator, the fine-tuned one can not only sort blur degrees properly but also reflect the distance between different motion blurs. A larger difference corresponds to a heavy motion blur of $\mathbf{I}_t$.

C. DeblurGAN-D for Selective Deblurring

After fine-tuning DeblurGAN-D, we remove its final mean operation and DeblurGAN-D becomes a fully-convolutional network. Hence, we can get a response map for an input patch or image. Each value of the response map indicates the blur degree of a region in the input. Fig. 10 shows the whole pipeline of our scheme.

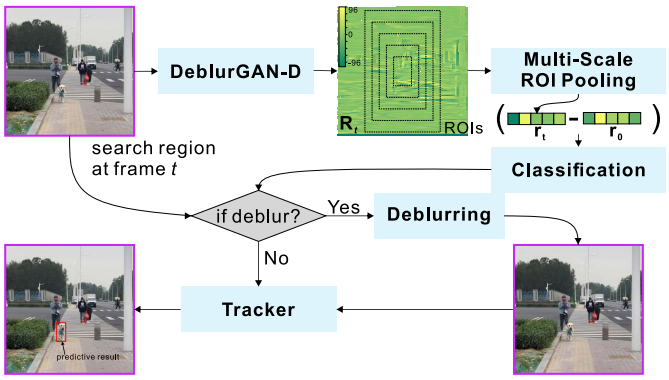

Fig. 10. The pipeline of our selective deblurring-based tracking. We can use existing deblurring methods, *e.g.*, DeblurGAN-G [14] for 'deblurring', and the classification is set as an offline trained SVM that indicates when we should deblurring a coming frame t.

TABLE I
COMPARISON RESULTS ON THE MOTION BLUR SUBSET OF OTB

Trackers	raw (AUC)	blur-robust tracking (AUC)
fDSST	0.512	**0.530**
Staple_CA	0.551	**0.561**
Siamfc	0.343	**0.353**
MBT	0.233	**0.242**
ECO	0.677	**0.679**
STRCF	0.633	**0.637**

In practice, given a tracker and an incoming frame t, we crop a search region and obtain a response map denoted as $\mathbf{R}_t$ by using DeblurGAN-D. Then, we conduct average ROI pooling at the center of $\mathbf{R}_t$ on 5 scales and get a vector $\mathbf{r}_t \in \Re^{5\times 1}$ that is further normalized by subtracting $\mathbf{r}_1$ that is the result of the first frame. We then use an offline trained SVM to do a binary classification by taking $\mathbf{r}_t - \mathbf{r}_1$ as the input. When SVM($\mathbf{r}_t - \mathbf{r}_1$)=1, it means that the search region is heavily-blurred and should be deblurred by DeblurGAN-G or SRN before following the tracking process, otherwise, the raw search region is used. Here, we use the DeblurGAN-G for deblurring.

We offline train the SVM via randomly selected 20 scenes of our dataset which corresponds to 80 videos with 4 levels of blur. We configure the selective ground truth as the selection results generated by '*_ganslt' in Section IV-B. We use a simple SVM to validate our scheme, which could be replaced with a more powerful classification, *e.g.*, deep convolutional neural network, in the future. We can equip extensive existing trackers with the proposed scheme.

D. Comparative Results

In the following, we equip 6 trackers, *i.e.*, STRCF [31], ECO [6], Siamfc [40], fDSST [75], Staple_CA [7], [8] and MBT [10], with the selective deblurring-based tracking scheme. we validate these improved trackers on our BVT benchmark and real motion blur subset of OTB benchmark [2].

1) BVT Benchmark Results: Since we have used 20 scenes, *i.e.*, 20×4 blurred videos, of our dataset to learn the SVM for selective deblurring in Section V-B, the remaining 80 scenes form new subsets denoted as $\{\mathcal{S}'^{L} | L = 2, 4, 8, 16\}$ each of which consists of 80 videos. We use them to validate the

TABLE II
TIME COSTS AND SPEEDS OF SIX TRACKERS AND THEIR BLUR-ROBUST VERISONS

Trackers	Ratio of selected frames for deblurring	Avg. resolution of search region	Avg. time per frame (ms)			Raw speed (fps)	Speed of our method (fps)
			Blur assessor	Deblurring	Tracking		
ECO	61.20%	338.89	4.85	26.29	284.10	3.52	3.17
STRCF	43.12%	333.56	5.29	25.47	22.17	45.10	18.90
fDSST	43.41%	230.54	3.28	12.17	14.64	68.31	33.24
Staple_CA	68.51%	340.84	3.27	26.60	13.18	75.89	23.23
Siamfc	33.61%	247.72	2.55	14.05	12.31	81.22	34.59
MBT	21.32%	139.44	2.90	4.45	25.90	38.61	30.07

proposed blur robust tracking scheme on the six representative trackers. We run the original six trackers and their improved versions on $\{\mathcal{S}'^{L}|L = 2, 4, 8, 16\}$ and calculate the average AUC gains on the whole dataset, *i.e.*, $\{\mathcal{S}'^{L}|L = 2, 4, 8, 16\}$, and its four subsets.

As shown in Fig. 11, according to the average AUC gains on $\{\mathcal{S}'^{L}|L = 2, 4, 8, 16\}$, all trackers are improved by the proposed blur-robust tracking scheme. In terms of average AUC gain on the four subsets with different blur levels, we observe that: 1) All trackers except STRCF and fDSST can get positive gains on lightly-blurred subsets. 2) the AUC gains of ECO, STRCF, fDSST, Staple_CA, and Siamfc gradually increase as motion blur becomes severer. In contrast, the AUC gains of MBT are larger on the light blur levels than on the severe ones. Such results are consistent with the observations in Section IV-B, that is, selective deblurring helps improve tracking accuracy and overcome the drawbacks of full deblurring, *i.e.*, leading to tracking accuracy decrease on lightly-blurred videos.

2) OTB-2015 Motion Blur Subset Results: We further compare the six trackers on the motion blur subset of OTB-2015. As shown in Table I, our method improves all six trackers' accuracy, which demonstrates that our method can be generalized to improve trackers on the real-blurred dataset. However, the above results do not mean the tracking performance is mainly determined by deblurring methods. As shown in Table I, although the tracking accuracies of six trackers are all improved through our framework, the ranks of all trackers are not changed, hinting that existing trackers' performance is mainly determined by their own properties, *e.g.*, the features and learning algorithms they used, instead of deblurring methods. For example, fDSST using HoG features could be enhanced by our method with the AUC on motion blur subset from 0.512 to 0.530, which still has a significant distance to the ECO tracker that uses more discriminative features and more advanced learning strategy. Note that, our work is to explore the effects of state-of-the-art deblurring methods to trackers and design an effective framework to limit the negative effects while encouraging the positive ones.

3) Time Cost and Complexity Analysis: The proposed framework employs two additional modules, *i.e.*, the blur assessor (DeblurGAN-D) and the deblurring method (DeblurGAN-G), inevitably leading to extra time costs. However, we argue that our method can maintain the real-time speed of existing trackers since only part of the frames are passed through the deblurring method and all extra calculations are conducted on search regions instead of the whole frames, which significantly limits the time costs. To make it clear, we decompose the whole process of a

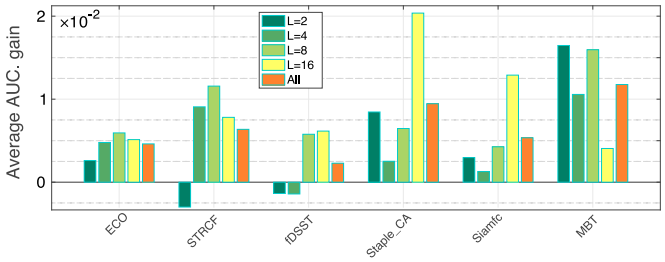

Fig. 11. Average AUC gains of six improved trackers over their original versions on the whole dataset, *i.e.*, $\{\mathcal{S}'^{L}|L = 2, 4, 8, 16\}$, and its four blurred subsets.

blur-robust tracker at each frame into three components, *i.e.*, the processes of blur assessing, deblurring, and tracking, and calculate the average per-frame time cost of each component on our BVT dataset. Note that, the time-consuming of the blur assessing and deblurring directly relates to the size of inputs and we use the search regions as their inputs. Besides, the deblurring method also depends on the ratio of selected frames needed to be deblurred. All trackers are one-by-one evaluated on the same computer with an Intel i7-8700 CPU and an NVIDIA RTX 2080 GPU. ECO is run in the CPU mode. We present the time costs and other related information of six enhanced trackers in Table II and observe that: *First*, the real-time trackers, *e.g.*, STRCF, fDSST, $\text{Staple}_{\text{CA}}$, Siamfc, and MBT, still remain real-time or near real-time speed (*i.e.*, the frame rate is around 25 fps). In particular, the speed of MBT method only decreases from 38.61 fps to 30.07 fps since the ratio of deblurred frames is low. *Second*, compared with the cost of the tracking process, the blur assessor only takes average 3.7 ms, having relatively mild effects on the tracking speed. In contrast, the deblurring method takes comparative time with the tracking process, affecting the final speed. The above results hint at more advanced and faster deblurring methods.

VI. CONCLUSION

In this paper, we proposed the Blurred Video Tracking (BVT) benchmark to explore how motion blur affects visual object tracking and whether state-of-the-art deblurring methods can benefit state-of-the-art trackers. The proposed BVT benchmark contains 500 videos for 100 scenes, each of which has 5 videos with different levels of motion blurs. According to the evaluation results of 25 trackers on the BVT benchmark, we find that light motion blur may have positive effects on visual tracking, while severe blurs certainly compromise the performance of most trackers. Using two state-of-the-art deblurring methods, DeblurGAN [14] and SRN [82], to deal with the blurred videos, we study the effects

of deblurring to 6 typical trackers. We observe that current deblurring algorithms can improve tracking performance on severely blurred videos while harming the accuracy of videos with light motion blur. Accordingly, we propose a general blur-robust tracking scheme that adopts a fine-tuned discriminator of DeblurGAN as an assessor to adaptively determine whether or not the current frame be deblurred. This method successfully improves the accuracy of 6 state-of-the-art trackers. We hope our observations could inspire the study of the blur robustness trackers while encouraging the development of real-time deblurring methods for visual object tracking.

REFERENCES

[1] Q. Guo *et al.*, "Watch out! Motion is blurring the vision of your deep neural networks," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, 2020.

[2] 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.

[3] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[4] 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.

[5] H. Fan *et al.*, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378.

[6] 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.

[7] 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. (CVPR)*, Jun. 2016, pp. 1401–1409.

[8] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1396–1404.

[9] J. Ding, Y. Huang, W. Liu, and K. Huang, "Severely blurred object tracking by learning deep image representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 319–331, Feb. 2016.

[10] B. Ma, L. Huang, J. Shen, L. Shao, M.-H. Yang, and F. Porikli, "Visual tracking under motion blur," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5867–5876, Dec. 2016.

[11] H. Jin, P. Favaro, and R. Cipolla, "Visual tracking in the presence of motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2005, pp. 18–25.

[12] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng, "Blurred target tracking by blur-driven tracker," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1100–1107.

[13] Y. Wu, J. Hu, F. Li, E. Cheng, J. Yu, and H. Ling, "Kernel-based motion-blurred target tracking," in *Advances in Visual Computing* (Lecture Notes in Computer Science), vol. 6939, G. Bebis *et al.*, Eds. Berlin, Germany: Springer, 2011, pp. 486–495, doi: 10.1007/978-3-642-24031-7_49.

[14] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192.

[15] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265.

[16] M. Noroozi, P. Chandramouli, and P. Favaro, "Motion deblurring in the wild," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 10496, V. Roth and T. Vetter, Eds. Cham, Switzerland: Springer, 2017, pp. 65–77, doi: 10.1007/978-3-319-66709-6_6.

[17] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 769–777.

[18] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1790–1798.

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

[20] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2015, pp. 564–586.

[21] M. Kristan *et al.*, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

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

[23] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan, "NUS-PRO: A new visual tracking challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 335–349, Feb. 2016.

[24] M. Kristan *et al.*, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Oct. 2017, pp. 1949–1972.

[25] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 310–327.

[26] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," 2018, *arXiv:1810.11981*. [Online]. Available: http://arxiv.org/abs/1810.11981

[27] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1134–1143.

[28] C. Seibold, A. Hilsmann, and P. Eisert, "Model-based motion blur estimation for the improvement of motion tracking," *Comput. Vis. Image Understand.*, vol. 160, pp. 45–56, Jul. 2017.

[29] S. Dai, M. Yang, Y. Wu, and A. K. Katsaggelos, "Tracking motion-blurred targets in video," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 2389–2392.

[30] C. Mei and I. Reid, "Modeling and generating complex motion blur for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[31] 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.

[32] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4847–4856.

[33] 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. 1144–1152.

[34] A. Lukezic *et al.*, "CDTB: A color and depth visual object tracking dataset and benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10012–10021.

[35] Y. Sun, C. Sun, D. Wang, Y. He, and H. Lu, "ROI pooled correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5776–5784.

[36] Q. Guo, W. Feng, C. Zhou, C.-M. Pun, and B. Wu, "Structure-regularized compressive tracking with online data-driven sampling," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5692–5705, Dec. 2017.

[37] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[38] Y. Song *et al.*, "VITAL: Visual tracking via adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8990–8999.

[39] I. Jung, J. Son, M. Baek, and B. Han, "Real-time mdnet," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 89–104.

[40] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," 2016, *arXiv:1606.09549*. [Online]. Available: http://arxiv.org/abs/1606.09549

[41] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1781–1789.

[42] Z. Zhu, Q. Wang, B. Li, W. Wei, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 103–119.

[43] X. Wang, C. Li, B. Luo, and J. Tang, "SINT++: Robust visual tracking via adversarial positive instance generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4864–4873.

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

[45] 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.

[46] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7944–7953.
[47] 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.
[48] 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.
[49] Z. Liang and J. Shen, "Local semantic siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 3351–3364, 2020.
[50] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention siamese network," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3068–3080, Jul. 2020.
[51] W. Feng, R. Han, Q. Guo, J. Zhu, and S. Wang, "Dynamic saliency-aware regularization for correlation filter-based object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3232–3245, Jul. 2019.
[52] Y. Yan *et al.*, "Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos," *Cognit. Comput.*, vol. 10, no. 1, pp. 94–104, Feb. 2018.
[53] Z. Wang, J. Ren, D. Zhang, M. Sun, and J. Jiang, "A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos," *Neurocomputing*, vol. 287, pp. 68–83, Apr. 2018.
[54] Y. Yan *et al.*, "Unsupervised image saliency detection with gestalt-laws guided optimization and visual attention based refinement," *Pattern Recognit.*, vol. 79, pp. 65–78, Jul. 2018.
[55] X. Dong, J. Shen, W. Wang, L. Shao, H. Ling, and F. Porikli, "Dynamical hyperparameter optimization via deep reinforcement learning in tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 29, 2019, doi: 10.1109/TPAMI.2019.2956703.
[56] T. Wang *et al.*, "Spatio-temporal point process for multiple object tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 8, 2020, doi: 10.1109/TNNLS.2020.2997006.
[57] J. Peng *et al.*, "TPM: Multiple object tracking with tracklet-plane matching," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107480.
[58] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[59] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875
[60] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," 2017, *arXiv:1704.00028*. [Online]. Available: http://arxiv.org/abs/1704.00028
[61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*. [Online]. Available: http://arxiv.org/abs/1606.03498
[62] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
[63] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890.
[64] B. Dolhansky and C. C. Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7902–7911.
[65] T. H. Kim, S. Nah, and K. M. Lee, "Dynamic video deblurring using a locally adaptive blur model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2374–2387, Oct. 2018.
[66] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6840–6848.
[67] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.
[68] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.
[69] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang, "Fast tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.
[70] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 36, 2012, pp. 864–877.
[71] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3101–3109.
[72] 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.
[73] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2014, pp. 254–265.
[74] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.
[75] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.
[76] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.
[77] 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. 4277–4286.
[78] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4591–4600.
[79] 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. 6181–6190.
[80] 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. 7181–7190.
[81] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," *ArXiv*, vol. abs/1608.07242, 2016. [Online]. Available: https://arxiv.org/abs/1608.07242
[82] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.
[83] Z. Chen, Q. Guo, L. Wan, and W. Feng, "Background-suppressed correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*. Piscataway, NJ, USA: IEEE Press, 2018, pp. 1–6.
[84] Q. Guo, R. Han, W. Feng, Z. Chen, and L. Wan, "Selective spatial regularization by reinforcement learned decision making for object tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 2999–3013, 2020.
[85] Q. Guo *et al.*, "SPARK: Spatial-aware online incremental attack against visual tracking," in *Proc. ECCV*, 2020.

Qing Guo (Member, IEEE) received the B.S. degree in electronic and information engineering from the North China Institute of Aerospace Engineering in 2011, the M.E. degree in computer application technology from the College of Computer and Information Technology, China Three Gorges University, in 2014, and the Ph.D. degree in computer application technology from the School of Computer Science and Technology, Tianjin University, China. He was a Research Fellow with the Nanyang Technology University, Singapore, from December 2019 to September 2020. He is currently a Wallenberg-NTU Presidential Postdoctoral Fellow with Nanyang Technology University. He is also with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University. His research interests include computer vision, AI security, and image processing.

Wei Feng (Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong in 2008. From 2008 to 2010, he was a Research Fellow with The Chinese University of Hong Kong and the City University of Hong Kong. He is currently a Full Professor with the School of Computer Science and Technology, College of Computing and Intelligence, Tianjin University, China. His research interests include active robotic vision and visual intelligence, specifically including active camera relocalization and lighting recurrence, general Markov Random Fields modeling, energy minimization, active 3D scene perception, SLAM, video analysis, and generic pattern recognition. Recently, he focuses on solving preventive conservation problems of cultural heritages via computer vision and machine learning. He is also the Associate Editor of *Neurocomputing* and *Journal of Ambient Intelligence and Humanized Computing*.

Ruijun Gao received the B.S. degree in computer science and technology from the College of Intelligence and Computing, Tianjin University, China, in 2015, where he is currently pursuing the M.S. degree. His research interests include computer vision and image processing.

Yang Liu (Senior Member, IEEE) received the B.Comp. degree (Hons.) from the National University of Singapore (NUS) in 2005 and the Ph.D. degree from NUS and MIT, in 2010. He started his postdoctoral work in NUS and MIT. In 2012, he joined Nanyang Technological University (NTU). He is currently a Full Professor and the Director of the Cybersecurity Laboratory, NTU. He specializes in software verification, security, and software engineering. His research has bridged the gap between the theory and practical usage of formal methods and program analysis to evaluate the design and implementation of software for high assurance and security. By now, he has more than 270 publications in top tier conferences and journals. He received a number of prestigious awards, including the MSRA Fellowship, the TRF Fellowship, the Nanyang Assistant Professor, the Tan Chin Tuan Fellowship, the Nanyang Research Award, and eight best paper awards in top conferences, such as ASE, FSE, and ICSE.

Song Wang (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 2002. He was a Research Assistant with Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning. He is also a Senior Member of the IEEE Computer Society. He is also serving as the Publicity/Web Portal Chair for the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society, and an Associate Editor for IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition Letters*, and *Electronics Letters*.