

AutoTrack: Towards High-Performance Visual Tracking for UAV with Automatic Spatio-Temporal Regularization

Yiming Li[†], Changhong Fu^{†,*}, Fangqiang Ding[†], Ziyuan Huang[‡], and Geng Lu[§]
[†]Tongji University [‡]National University of Singapore [§]Tsinghua University
 yimingli9702@gmail.com, changhongfu@tongji.edu.cn, lug@tsinghua.edu.cn

Abstract

Most existing trackers based on discriminative correlation filters (DCF) try to introduce predefined regularization term to improve the learning of target objects, e.g., by suppressing background learning or by restricting change rate of correlation filters. However, predefined parameters introduce much effort in tuning them and they still fail to adapt to new situations that the designer did not think of. In this work, a novel approach is proposed to online automatically and adaptively learn spatio-temporal regularization term. Spatially local response map variation is introduced as spatial regularization to make DCF focus on the learning of trust-worthy parts of the object, and global response map variation determines the updating rate of the filter. Extensive experiments on four UAV benchmarks have proven the superiority of our method compared to the state-of-the-art CPU- and GPU-based trackers, with a speed of ~ 60 frames per second running on a single CPU.

Our tracker is additionally proposed to be applied in UAV localization. Considerable tests in the indoor practical scenarios have proven the effectiveness and versatility of our localization method. The code is available at <https://github.com/vision4robotics/AutoTrack>.

1. Introduction

Visual object tracking is one of the fundamental tasks in the computer vision community, aiming to localize the object sequentially only with the information given in the first frame. Endowing unmanned aerial vehicle (UAV) with visual tracking capability brings many applications, e.g., aerial cinematography [1], person following [2], aircraft tracking [3], and traffic patrolling [4].

There are currently two main research interests in this area: discriminative correlation filter (DCF)-based methods [5, 6, 7, 8, 9, 10, 11, 12, 13, 14] as well as deep learning-based approaches [15, 16, 17, 18, 19, 20]. In considera-

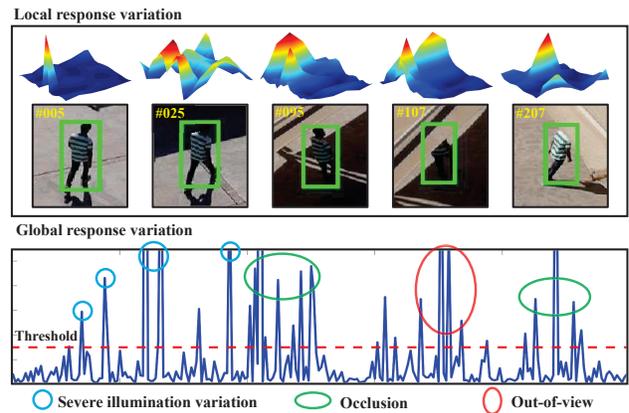


Figure 1. Central idea of our tracker. Spatially local and global response variations are exploited. Local variations indicate local credibility in the object bounding box. Severe illumination change in frame 25 and 95 as well as partial occlusion in frame 107 and 207 can lower the credibility of the appearance. AutoTrack is punished for learning these appearances so that local distractions can be avoided. In terms of global variations, large value can indicate wrong tracking result, where we stop the learning of correlation filters, while relatively large value should accelerate the learning of correlation filters so that adaptivity can be raised.

tion of the limitation of power capacity and computational resources onboard UAVs, DCF framework is selected because of its high efficiency originating from calculation in the Fourier domain.

To improve DCF-based trackers, there are currently three directions: a) building more robust appearance model [18, 17, 20, 21], b) mitigating boundary effect or imposing restrictions in learning [8, 22, 17, 14, 23], and c) mitigating filter degradation [24, 12, 14, 25]. Robust appearance can indeed boost performance, yet it leads to burdensome calculations. Filter degradation, on the other hand, is not improving it fundamentally. Most trackers try to improve performance using option b) by introducing regularization terms.

Recently, some attentions have been brought to using response maps generated in the detection phase to form the restrictions in learning [26]. The intuition behind it is

*Corresponding author

that the response map contains crucial information regarding the resemblance of current object and the appearance model. However, [26] only exploits what we call the spatially global response map variations, while ignoring local response variation indicating credibility at different locations in the image: drastic local variation means low credibility and vice versa.

We fully exploit the local-global response variation to train our tracker with automatic spatio-temporal regularization, *i.e.*, AutoTrack. While most parameters in regularization terms proposed by others are hyper-parameters that require large effort to tune, and would have a difficult time adjusting to new situations that the designers did not think of, we propose to learn some of the hyper-parameters automatically and adaptively. AutoTrack performs favorably against the state-of-the-art trackers, while running at ~ 60 frames per second (fps) on a single CPU.

Our main contributions are summarized as follows:

- We propose a novel spatio-temporal regularization term to simultaneously exploit local and global information hidden in response maps.
- We develop a novel DCF-based tracker which can automatically tune the hyper-parameters of spatio-temporal regularization term on the fly.
- We evaluate our tracker on 278 difficult UAV image sequences, and the evaluations have validated the state-of-the-art performance of our tracker compared to current CPU- and GPU-based trackers.
- We introduce a novel application of visual object tracking in UAV localization and prove its effectiveness as well as generality in the practical scenarios.

2. Related Works

Tracking by detection: tracking-by-detection framework, which regards the tracking as a classification problem, is widely adopted in UAV [5, 6, 7, 27, 28]. Among them, DCF has exhibited good performance with exceptional efficiency. The speed of traditional DCF-based trackers [7, 5, 29] is around hundreds of fps on a single CPU, far exceeding the real-time requirement of UAV (30 fps). Yet they are primarily subjected to the following issues.

a) Boundary effect: the circulant samples suffer from periodical splicing at the boundary, reducing filters' discriminative power. Several works can mitigate boundary effect [22, 8, 14, 30], but they used a constant spatial penalization which cannot adapt to various changes in different objects. K. Dai *et al.* optimized the spatial regularization in the temporal domain [31]. Different to [31], we exploit the inherent information in DCF framework, so our method is more generic. Also, we have achieved better performance in the aerial scenarios in terms of speed and precision.

b) Filter degradation: the appearance model updated via a linear interpolation method cannot adapt to ubiquitous appearance change, leading to filter degradation. Some attempts are made to tackle the issue, *e.g.*, training set management [20, 24, 32], temporal restriction [14, 25], tracking confidence verification [23, 12] and over-fitting alleviation [33]. Amongst them the temporal regularization is an effective and efficient way. Yet the non-adaptive regularization is prone to tracking drift once the filter is corrupted.

Tracking by deep learning: recently, deep learning-based tracking has caught wide attention due to its robustness, *e.g.*, deep feature representation [18, 34, 17, 20], reinforcement learning [16], residual learning [35] and adversarial learning [36]. However, for mobile robots, the above trackers cannot meet the requirement of real-time perception even with a high-end GPU. Currently, the state-of-the-art deep trackers [14, 37, 38, 39, 40, 41] are mostly built on siamese neural network [15]. The pre-trained siamese trackers just need to traverse in a feed-forward way to get a similarity score for object localization, facilitating real-time implementation on GPU. However, on a mobile device solely with CPU, the speed of siamese-based trackers cannot satisfy the real-time needs. C. Huang *et al.* proposed a CPU-friendly deep tracker [42] by training an agent working in a cascaded manner. It can run at near real-time speed by reducing calculation on easy frames. In summary, deep trackers can hardly meet real-time demands on CPU.

Vision-based localization: vision-based localization is crucial for UAV especially in GPS-denied environments. A. Breitenmoser *et al.* developed a monocular 6D pose estimation system based on passive markers in the visible spectrum [43]. However, it performs worse in low-light environments. M. Faessler *et al.* presented a monocular localization system based on infrared LEDs to raise robustness in cluttered environments [44]. Its generality, however, is limited since the system can only work in the infrared spectrum. Built on [44], we develop a localization system based on visual tracking. In light of robustness and generality of our tracker in various scenarios like illumination variation, occlusion and deformation, our localization system is more versatile compared to the infrared LED-based one [44].

3. Revisit STRCF

In this section, our baseline STRCF [14] is revisited. The optimal filter \mathbf{H}_t in frame t is learned by minimizing the following objective function:

$$\begin{aligned} \mathcal{E}(\mathbf{H}_t) = & \frac{1}{2} \|\mathbf{y} - \sum_{k=1}^K \mathbf{x}_t^k \otimes \mathbf{h}_t^k\|_2^2 + \frac{1}{2} \sum_{k=1}^K \|\mathbf{u} \odot \mathbf{h}_t^k\|_2^2 \\ & + \frac{\theta}{2} \sum_{k=1}^K \|\mathbf{h}_t^k - \mathbf{h}_{t-1}^k\|_2^2 \end{aligned} \quad (1)$$

where $\mathbf{x}_t^k \in \mathbb{R}^{T \times 1}$ ($k = 1, 2, 3, \dots, K$) is the extracted feature with length T in frame t , and K denotes number of

channel, $\mathbf{y} \in \mathbb{R}^{T \times 1}$ is the desired Gaussian-shaped response. $\mathbf{h}_t^k, \mathbf{h}_{t-1}^k \in \mathbb{R}^{T \times 1}$ respectively denote the filter of the k -th channel trained in the t -th and $(t-1)$ -th frame, \otimes indicates the convolution operator. Noted that $\mathbf{H}_t = [\mathbf{h}_t^1, \mathbf{h}_t^2, \mathbf{h}_t^3, \dots, \mathbf{h}_t^K]$. As for regularization, the spatial regularization parameter $\mathbf{u} \in \mathbb{R}^{T \times 1}$ is bowl-shaped and borrowed from SRDCF [8] for decreasing boundary effect, and temporal regularization, *i.e.*, the third term in Eq. 1, is firstly proposed to restrict filter's variation by penalizing the difference between the current and previous filters.

Although STRCF [14] has achieved competent performance, it does have two limitations: a) the fixed spatial regularization failing to address appearance variation in the unforeseeable aerial tracking scenarios, b) the unchanged temporal penalty strength θ (set as 15 in [14]) which is not general in all kinds of situations.

4. Automatic Spatio-Temporal Regularization

In this work, both local and global response variations are fully utilized to achieve simultaneous spatial and temporal regularizations, as well as automatic and adaptive hyperparameter optimization.

4.1. Response Variation

First of all, we define local response variation vector $\mathbf{\Pi} = [|\Pi^1|, |\Pi^2|, \dots, |\Pi^T|]$, as can be seen in Fig. 1 for its 2D visualization in the object bounding box, in preparation for spatial regularization. Its i -th element $|\Pi^i|$ is defined as:

$$\Pi^i = \frac{\mathcal{R}_t[\psi_\Delta]^i - \mathcal{R}_{t-1}^i}{\mathcal{R}_{t-1}^i}, \quad (2)$$

where $[\psi_\Delta]$ is the shift operator to make two peaks in two response maps \mathcal{R}_t and \mathcal{R}_{t-1} coincide with each other, in order for removing the motion influence [26]. \mathcal{R}^i denotes the i -th element in response map \mathcal{R} .

Automatic spatial regularization: local response variation reveals the credibility of every pixel in the search area of the current frame. Therefore, filters located where the pixel credibility is low should be restricted in learning. We achieve this by introducing local variation $\mathbf{\Pi}$ to the spatial regularization parameter $\tilde{\mathbf{u}}$:

$$\tilde{\mathbf{u}} = \mathbf{P}^\top \delta \log(\mathbf{\Pi} + 1) + \mathbf{u}, \quad (3)$$

where $\mathbf{P}^\top \in \mathbb{R}^{T \times T}$ is used to crop the central part of the filter where the object is located. δ is a constant to adjust the weight of local response variations, and \mathbf{u} is inherited from STRCF [14] to mitigate boundary effects. Through Eq. 3, filters located at pixels with dramatic response variation will be partially refrained from learning the new appearance because of the spatial punishment.

Automatic temporal regularization: in STRCF [14], the change rate of filters between two frames is punished in

the loss by a fixed parameter θ . AutoTrack tries to adaptively and automatically determine the value of this hyperparameter by jointly optimization of its value and the filter. So we define a reference $\tilde{\theta}$ in preparation for the objective function with regard to the global response:

$$\tilde{\theta} = \frac{\zeta}{1 + \log(\nu \|\mathbf{\Pi}\|_2 + 1)}, \quad \|\mathbf{\Pi}\|_2 \leq \phi, \quad (4)$$

where ζ and ν denote hyper parameters. When the global variation is higher than the threshold ϕ , it means that there are aberrances in response maps [26], so correlation filter ceases to learn. If it is lower than the threshold, the more dramatic the response map varies, the smaller the reference value will be, so that the restriction on temporal change of the correlation filters can be loosened and it can learn more rapidly in situations like large appearance variations.

Remark 1: Note that what we defined here is the reference value rather than the hyper-parameter itself. For the hyper-parameter of the temporal regularization, we use joint optimization to online estimate the value of it, so that the restriction can be online adaptively adjusted according to the response map variations. When appearance changes drastically, correlation filter learns more rapidly and vice versa.

4.2. Objective Optimization

Our objective function for joint optimization of filter as well as temporal regularization term can be written as:

$$\begin{aligned} \mathcal{E}(\mathbf{H}_t, \theta_t) = & \frac{1}{2} \|\mathbf{y} - \sum_{k=1}^K \mathbf{x}_t^k \otimes \mathbf{h}_t^k\|_2^2 + \frac{1}{2} \sum_{k=1}^K \|\tilde{\mathbf{u}} \odot \mathbf{h}_t^k\|_2^2 \\ & + \frac{\theta_t}{2} \sum_{k=1}^K \|\mathbf{h}_t^k - \mathbf{h}_{t-1}^k\|_2^2 + \frac{1}{2} \|\theta_t - \tilde{\theta}\|_2^2 \end{aligned}, \quad (5)$$

where $\tilde{\theta}$ and θ_t respectively denote the reference and optimized temporal regularization parameter, and $\tilde{\mathbf{u}}$ represents the automatic spatial regularization calculated via Eq. 3.

For optimization, we introduce an auxiliary variable $\hat{\mathbf{g}}_t$ by ordering $\hat{\mathbf{g}}_t = \sqrt{T} \mathbf{F} \mathbf{h}_t$ ($\hat{\mathbf{G}} = [\hat{\mathbf{g}}_t^1, \hat{\mathbf{g}}_t^2, \hat{\mathbf{g}}_t^3, \dots, \hat{\mathbf{g}}_t^K]$) where $\mathbf{F} \in \mathbb{C}^{T \times T}$ denotes the orthonormal matrix and the symbol $\hat{\cdot}$ denotes the discrete Fourier transform (DFT) of a signal. Then Eq. 5 is converted into the frequency domain:

$$\begin{aligned} \mathcal{E}(\mathbf{H}_t, \theta_t, \hat{\mathbf{G}}_t) = & \frac{1}{2} \|\mathbf{y} - \sum_{k=1}^K \hat{\mathbf{x}}_t^k \odot \hat{\mathbf{g}}_t^k\|_2^2 + \frac{1}{2} \sum_{k=1}^K \|\tilde{\mathbf{u}} \odot \mathbf{h}_t^k\|_2^2 \\ & + \frac{\theta_t}{2} \sum_{k=1}^K \|\hat{\mathbf{g}}_t^k - \hat{\mathbf{g}}_{t-1}^k\|_2^2 + \frac{1}{2} \|\theta_t - \tilde{\theta}\|_2^2 \end{aligned}. \quad (6)$$

By minimizing Eq. 6, an optimal solution can be obtained through alternating direction method of multipliers (ADMM) [45]. The Augmented Lagrangian form of equa-

tion Eq. 6 can be formulated as:

$$\begin{aligned} \mathcal{L}_t(\mathbf{H}_t, \theta_t, \widehat{\mathbf{G}}_t, \widehat{\mathbf{M}}_t) &= \mathcal{E}(\mathbf{H}_t, \theta_t, \widehat{\mathbf{G}}_t) + \frac{\gamma}{2} \sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \sqrt{T}\mathbf{F}\mathbf{h}_t^k\|_2^2 \\ &+ \sum_{k=1}^K (\widehat{\mathbf{g}}_t^k - \sqrt{T}\mathbf{F}\mathbf{h}_t^k)^\top \widehat{\mathbf{m}}_t^k \end{aligned} \quad (7)$$

where $\widehat{\mathbf{M}}_t = [\widehat{\mathbf{m}}_1, \widehat{\mathbf{m}}_2, \dots, \widehat{\mathbf{m}}_K] \in \mathbb{R}^{T \times K}$ is the Fourier transform of the Lagrange multiplier and γ denotes the step size regularization parameter. By assigning $\mathbf{v}_t^k = \frac{\mathbf{m}_t^k}{\gamma}$ ($\mathbf{V}_t^k = [\mathbf{v}_t^1, \mathbf{v}_t^2, \dots, \mathbf{v}_t^K]$), Eq. 7 can be reformulated as:

$$\begin{aligned} \mathcal{L}_t(\mathbf{H}_t, \theta_t, \widehat{\mathbf{G}}_t, \widehat{\mathbf{V}}_t) &= \mathcal{E}(\mathbf{H}_t, \theta_t, \widehat{\mathbf{G}}_t) \\ &+ \frac{\gamma}{2} \sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \sqrt{T}\mathbf{F}\mathbf{h}_t^k + \widehat{\mathbf{v}}_t^k\|_2^2. \end{aligned} \quad (8)$$

Then we solve the following subproblems by ADMM.

Subproblem $\widehat{\mathbf{G}}$: given $\mathbf{H}_t, \theta_t, \widehat{\mathbf{V}}_t$, the optimal $\widehat{\mathbf{G}}^*$ is:

$$\begin{aligned} \widehat{\mathbf{G}}^* &= \arg \min_{\widehat{\mathbf{G}}} \left\{ \frac{1}{2} \|\widehat{\mathbf{y}} - \sum_{k=1}^K \widehat{\mathbf{x}}_t^k \odot \widehat{\mathbf{g}}_t^k\|_2^2 \right. \\ &+ \frac{\theta_t}{2} \sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \widehat{\mathbf{g}}_{t-1}^k\|_2^2 + \frac{\gamma}{2} \sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \sqrt{T}\mathbf{F}\mathbf{h}_t^k + \widehat{\mathbf{v}}_t^k\|_2^2 \left. \right\}. \end{aligned} \quad (9)$$

Solving Eq. 9 directly is very difficult because of its complexity. So we decide to sample $\widehat{\mathbf{x}}_t$ across all K channels in each pixel to simplify our formulation written by:

$$\begin{aligned} \Gamma_j^*(\widehat{\mathbf{G}}_t) &= \arg \min_{\Gamma_j(\widehat{\mathbf{G}}_t)} \{ \|\widehat{\mathbf{y}}_j - \Gamma_j(\widehat{\mathbf{X}}_t)^\top \Gamma_j(\widehat{\mathbf{G}}_t)\|_2^2 \\ &+ \gamma \|\Gamma_j(\widehat{\mathbf{G}}_t) + \Gamma_j(\widehat{\mathbf{V}}_t) - \Gamma_j(\sqrt{T}\mathbf{F}\mathbf{H}_t)\|_2^2, \\ &+ \theta_t \|\Gamma_j(\widehat{\mathbf{G}}_t) - \Gamma_j(\widehat{\mathbf{G}}_{t-1})\|_2^2 \} \end{aligned} \quad (10)$$

where $\Gamma_j(\widehat{\mathbf{X}}) \in \mathbb{C}^{K \times 1}$ represents the vector containing values of all K channels of $\widehat{\mathbf{X}}$ on pixel j ($j = 1, 2, \dots, T$). After derivation using Sherman Morrison formula, we can obtain its solution:

$$\Gamma_j^*(\widehat{\mathbf{G}}_t) = \frac{1}{\gamma + \theta_t} \left(\mathbf{I} - \frac{\Gamma_j(\widehat{\mathbf{X}}_t) \Gamma_j(\widehat{\mathbf{X}}_t)^\top}{\theta_t + \gamma + \Gamma_j(\widehat{\mathbf{X}}_t)^\top \nu_j(\widehat{\mathbf{X}}_t)} \right) \boldsymbol{\rho}, \quad (11)$$

where the vector $\boldsymbol{\rho}$ takes the form $\boldsymbol{\rho} = \Gamma_j(\widehat{\mathbf{X}}_t) \widehat{\mathbf{y}}_j + \theta_t \Gamma_j(\widehat{\mathbf{G}}_{t-1}) - \gamma \Gamma_j(\widehat{\mathbf{V}}_t) + \gamma \Gamma_j(\sqrt{T}\mathbf{F}\mathbf{H}_t)$ for presentation.

Subproblem \mathbf{H} : given $\theta_t, \widehat{\mathbf{G}}_t, \widehat{\mathbf{V}}_t$, we can optimize \mathbf{h}^k by:

$$\arg \min_{\mathbf{h}^k} \left\{ \frac{1}{2} \|\widehat{\mathbf{u}} \odot \mathbf{h}_t^k\|_2^2 + \frac{\gamma}{2} \|\widehat{\mathbf{g}}_t^k - \sqrt{T}\mathbf{F}\mathbf{h}_t^k + \widehat{\mathbf{v}}_t^k\|_2^2 \right\}. \quad (12)$$

The closed-form solution of \mathbf{h}^k can be written by:

$$\mathbf{h}^{k*} = [\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{U}} + \gamma T \mathbf{I}]^{-1} \gamma T (\mathbf{v}_t^k + \mathbf{g}_t^k) = \frac{\gamma T (\mathbf{v}_t^k + \mathbf{g}_t^k)}{(\widetilde{\mathbf{u}} \odot \widetilde{\mathbf{u}}) + \gamma T}, \quad (13)$$

where $\widetilde{\mathbf{U}} = \text{diag}(\widetilde{\mathbf{u}}) \in \mathbb{R}^{T \times T}$ represents diagonal matrix.

Subproblem θ_t : given other variables in Eq. 8, the optimal solution of θ_t can be determined as:

$$\begin{aligned} \theta_t^* &= \arg \min_{\theta_t} \left\{ \frac{\theta}{2} \sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \widehat{\mathbf{g}}_{t-1}^k\|_2^2 + \frac{1}{2} \|\theta_t - \tilde{\theta}\|_2^2 \right\} \\ &= \tilde{\theta} - \frac{\sum_{k=1}^K \|\widehat{\mathbf{g}}_t^k - \widehat{\mathbf{g}}_{t-1}^k\|_2^2}{2} \end{aligned} \quad (14)$$

Lagrangian multiplier update: after solving three subproblems above, we can update Lagrangian multipliers as:

$$\widehat{\mathbf{V}}^{i+1} = \widehat{\mathbf{V}}^i + \gamma^i (\widehat{\mathbf{G}}^{i+1} - \widehat{\mathbf{H}}^{i+1}), \quad (15)$$

where i and $i + 1$ denotes the iteration index and the step size regularization constant γ (initially equals to 1) takes the form of $\gamma^{(i+1)} = \min(\gamma_{max}, \beta \gamma^i)$. ($\beta = 10, \gamma_{max} = 10000$)

By iteratively solving the four subproblems above, we can optimize our objective function effectively and obtain the optimal filter $\widehat{\mathbf{G}}_t$ and temporal regularization parameter θ_t in frame t . Then $\widehat{\mathbf{G}}_t$ is used for detection in frame $t + 1$.

4.3. Object Localization

The tracked object is localized by searching for the maximum value of response map \mathcal{R}_t calculated by:

$$\mathcal{R}_t = \mathcal{F}^{-1} \sum_{k=1}^K (\widehat{\mathbf{z}}_t^k \odot \widehat{\mathbf{g}}_{t-1}^k), \quad (16)$$

where \mathcal{R}_t is the response map in frame t , \mathcal{F}^{-1} denotes the inverse Fourier transform (IFT) operator and $\widehat{\mathbf{z}}_t^k$ represents the Fourier form of extracted feature map in frame t .

5. Localization by Tracking

Self-localization for UAV is essential for autonomous navigation. To develop a robust and universal localization system in dynamic and uncertain environments, we introduce visual object tracking into UAV localization for the first time. Specifically, we utilize the open-source software in [44], but employ AutoTrack to track four objects simultaneously instead of segmenting LEDs in the infrared spectrum. The main work-flow is briefly described below.

Prerequisites: the system requires the knowledge of four object configuration (non-symmetric), *i.e.*, their positions in the world coordinate (observed in motion capture system), and intrinsic UAV-mounted camera parameters.

Initialization and tracking: after manually assigning four objects, AutoTrack starts to track them independently and output their location in the RGB image. Different to the system [44] only applicable in infrared spectrum, our system can be used in versatile environments.

Correspondence search and pose optimization: correspondence between the tracked object configuration in the world coordinate and tracked results in image frames is firstly clarified, then the final 6D pose is optimized by fine-tuning the reprojection error [44].

6. Experiments

In this section, we firstly evaluate the tracking performance of AutoTrack with current state-of-the-art trackers on four difficult UAV benchmarks [46, 47, 48, 49]. Then, the proposed localization system is evaluated on Quanser* platform in the indoor practical scenarios. The experiments of tracking performance evaluation are conducted using MATLAB R2018a on a PC with an i7-8700K processor (3.7GHz), 32GB RAM and NVIDIA GTX 2080 GPU. The tests of localization system are run on ROS [50] using C++. For the hyper parameters of AutoTrack, we set $\delta = 0.2$, $\nu = 2 \times 10^{-5}$, $\zeta = 13$. The threshold of ϕ is 3000, ADMM iteration is set to 4. The sensitivity analysis of all the parameters can be found in the supplementary material.

6.1. Evaluation on UAV Datasets

For rigorous and comprehensive evaluation, the comparison between AutoTrack with the state-of-the-art methods is reported on four challenging and authoritative UAV benchmarks: DTB70 [46], UAVDT [47], UAV123@10fps [48] and VisDrone2018-test-dev [49], with a total number of 119,830 frames. Noted that we use the same evaluation criteria with the four benchmarks [46, 47, 48, 49].

6.1.1 Comparison with deep-based trackers

DTB70: DTB70 [46], composed of 70 difficult UAV image sequences, primarily addresses the problem of severe UAV motion. In addition, various cluttered scenes and objects with different sizes as well as aspect ratios are included. We compare AutoTrack with nine state-of-the-art deep trackers, *i.e.*, ASRCF [31], TADT [51], HCF [18], ADNet [16], CFNet [52], UDT+ [53], IBCCF [54], MDNet [55], MCPF [19], on DTB70, and the final results are reported in Fig. 2. Only with hand-crafted features, AutoTrack outperforms deep feature-based trackers (ASRCF [31], HCF [18], MCPF [19] and IBCCF [54]) and pre-trained deep architecture-based trackers, *i.e.*, MDNet [55], ADNet [16], UDT+ [53] and CFNet [52]. In summary, AutoTrack exhibits strong robustness against drastic UAV motion without losing efficiency, and also demonstrates a generality in tracking different objects in various scenes.

UAVDT: UAVDT [47] mainly emphasizes vehicle tracking in various scenarios. Weather condition, flying altitude and camera view are three categories addressed by UAVDT. Compared to deep trackers including ASRCF [31], TADT [51], SiameseFC [15], DSiam [56], MCCT [13], ADNet [16], CFNet [52], DeepSTRCF [14], UDT+ [53], HCF [18], C-COT [17], ECO [20], IBCCF [54], MCPF [19] and CREST [35], AutoTrack with a single CPU exhibits the

*<https://www.quanser.com/products/autonomous-vehicles-research-studio/>

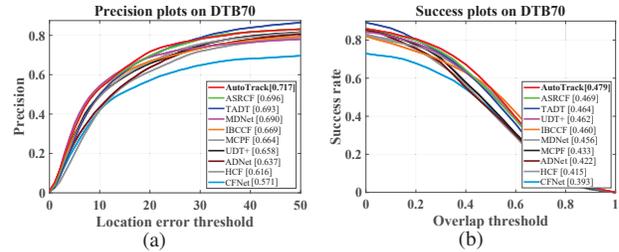


Figure 2. Overall comparison with deep trackers on DTB70 [46]. AutoTrack ranks first place in both precision and success rate.

best performance in terms of precision and speed, as shown in Table 1. In a word, AutoTrack has extraordinary performance in vehicle tracking despite omnipresent challenges.

6.1.2 Comparison with CPU-based trackers

Twelve real-time trackers (with a speed of >30 fps), *i.e.*, KCF [7], DCF [7], KCC [57] fDSST [11], DSST [58], BACF [22], STAPLE-CA [10], STAPLE [10], MCCT-H [13], STRCF [14], ECO-HC [20], ARCF-H [26], and five non-real-time ones, *i.e.*, SRDCF [8], SAMF [6], CSRDCF [30], SRDCFdecon [24], ARCF-HC [26] are used for comparison. The results of real-time trackers on four datasets are displayed in Fig. 3. Besides, the average performance of top ten CPU-based trackers in terms of speed and precision is demonstrated in the Table 2. It can be seen that AutoTrack is the best real-time tracker on CPU. Some tracking results are demonstrated in Fig. 4 and Fig. 6.

Overall performance evaluation: AutoTrack has outperformed all the CPU-based real-time trackers in both precision and success rate on DTB70 [46], UAVDT [47] and UAV123@10fps [48]. On VisDrone2018-test-dev [49], AutoTrack achieves comparable performance with the best tracker MCCT-H and ECO-HC in terms of precision and success rate. As for the average performance of top ten

Table 1. Precision and speed comparison between AutoTrack with deep trackers on UAVDT [47]. * means GPU speed. Red, green and blue respectively mean the first, second and third place.

Tracker	Precision	FPS	Tracker	Precision	FPS
AutoTrack	71.8	65.4	UDT+[53]	69.7	60.4*
DeepSTRCF[14]	66.7	6.6*	ADNet[16]	68.3	7.6*
DSiam[56]	70.4	15.9*	TADT[51]	67.7	32.5*
MCPF[19]	66.0	0.67*	MCCT[13]	67.1	8.6*
Siamese[15]	68.1	37.9*	ECO[20]	70.0	16.4*
C-COT[17]	65.6	1.1*	CREST[35]	64.9	4.3*
ASRCF[31]	70.0	24.1*	HCF[18]	60.2	20.15*
CFNet[52]	68.0	41.1*	IBCCF[54]	60.3	3.39*

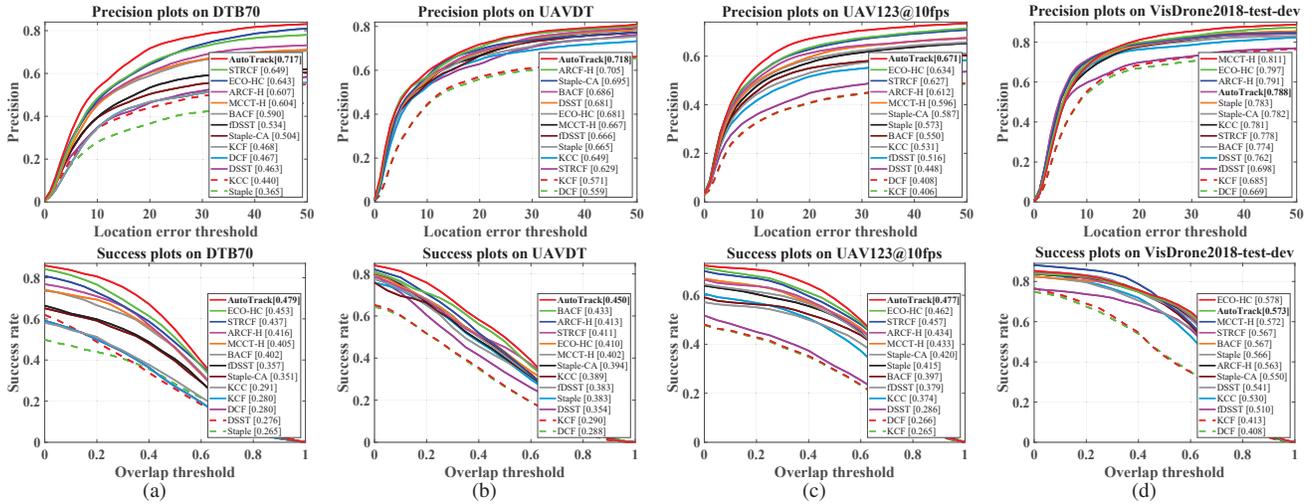


Figure 3. Overall performance of CPU-based real-time trackers on (a) DTB70 [46] (b) UAVDT [47] (c) UAV123@10fps [48] and (d) VisDrone2018-test-dev [49]. Two measures for one-pass evaluation (OPE) [59] are used for evaluation. Precision plot can demonstrate the percentage of scenarios when the distance between estimated bounding box and ground truth one is smaller than different thresholds, and the score at 20 pixels is used for ranking. Success plot can display the percentage of situations when the overlap between estimated bounding box and ground truth one is greater than different thresholds. Area under curve (AUC) is utilized for ranking.

Table 2. Average speed (fps) and precision of top ten CPU-based trackers on four benchmarks. Red, green and blue respectively mean the first, second and third place. All the reported speed is run on a single CPU. Noted that AutoTrack is the best real-time tracker on CPU.

Tracker	AutoTrack	ARCF-HC[26]	ECO-HC[20]	ARCF-H[26]	STRCF[14]	MCCT-H[13]	STAPLE_CA[10]	BACF[22]	CSR-DCF[30]	SRDCF[8]
Precision	72.4	71.9	69.1	67.3	67.1	67.0	64.2	65.6	67.7	62.7
Speed	59.2	19.3	69.5	53.4	28.4	58.8	58.5	53.1	11.8	14.2

CPU-based trackers, AutoTrack has the best performance in precision with the second fast speed of 59.2fps, only slower than ECO-HC (69.5fps), however, we have achieved an average improvement of 4.8% in precision compared to ECO-HC. Moreover, AutoTrack has an advantage of 7.9% in precision and 108.5% in speed over the baseline STRCF.

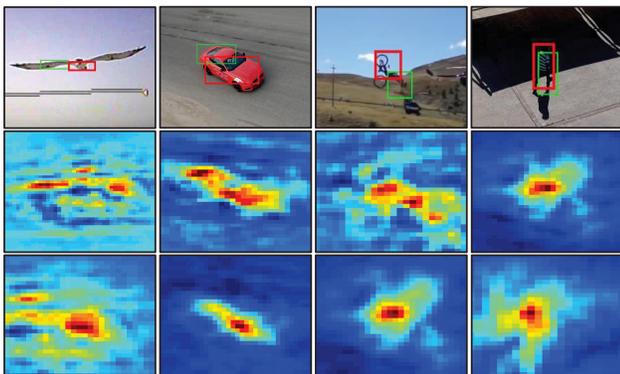


Figure 4. Tracking results and response maps of AutoTrack (red box) and STRCF (green box) of *bird1_3*, *car18*, *MountainBike5* and *person12_2*. AutoTrack (third row) has less distraction in response than STRCF (second row) due to automatic regularization.

Remark 2: M. Muller *et al.* created a 10fps dataset from the recorded 30fps one [48], thus the movement of tracked object between successive frames is larger, bringing more challenges. On UAV123@10fps, AutoTrack achieves a remarkable advantage of 5.8% in precision than the second best ECO-HC, proving its robustness against large motion.

Remark 3: Compared to ARCF-HC [26] solely repressing the global response variation using a fixed parameter, we fully utilize the local-global information to fine-tune the spatio-temporal regularization term in an automatic manner. Extensive experiments have shown that AutoTrack achieves better performance while providing a much faster speed which is 3.1 times that of ARCF-HC.

Attribute-based evaluation: Success plots of eight attributes are exhibited in Fig. 5. In the normal appearance change scenarios (deformation, in-plane-rotation, viewpoint change), AutoTrack improves STRCF by 15.9%, 15.5% and 4.6% in success rate because the automatic temporal regularization can smoothly help filter adapt to new appearance. In illumination variation and large occlusion (aberrant appearance variation), AutoTrack has a superiority of 7.0% and 15.7% compared to STRCF in light of adaptive spatial regularization as well as aberrance monitoring mechanism which can stop training before contamination.

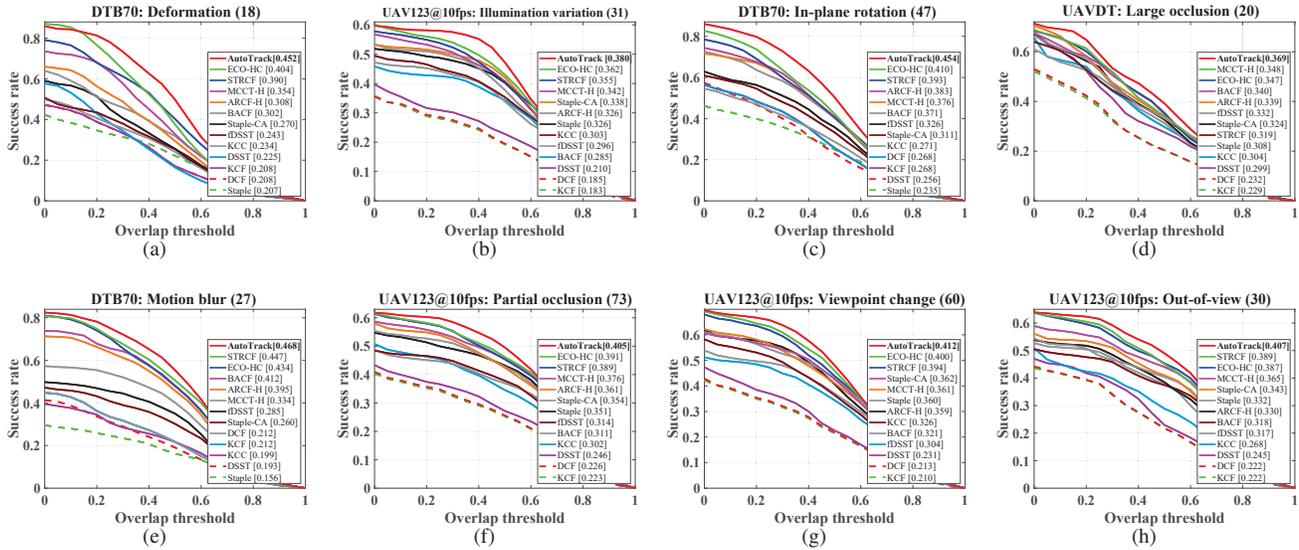


Figure 5. Attribute-based comparison on deformation, illumination variation, in-plane rotation, large occlusion, motion blur, partial occlusion, viewpoint change, and out-of-view. More attribute-based evaluations can be seen in the supplementary material.

6.1.3 Ablation study

To validate the effectiveness of our method, AutoTrack is compared to itself with different modules enabled. The overall evaluation is presented in Table 3. With each module (automatic spatial regularization ASR, automatic temporal regularization ATR) added to the STRCF, the performance is smoothly improved. It is noted that ATR can also bring a gain in speed compared to ASR because we can reduce meaningless and detrimental training on contaminated samples. In addition, response maps of some frames are illustrated in Fig. 4. It can be clearly seen that response of our method is more reliable than that of baseline.

Table 3. Ablation study of AutoTrack. ASR and ATR respectively represents automatic spatial and temporal regularization.

Tracker	Precision	AUC	FPS
STRCF	0.671	0.468	28.4
STRCF + ASR	0.716	0.489	53.7
STRCF + ATR	0.714	0.492	60.0
AutoTrack	0.724	0.495	59.2

6.2. Evaluation of Localization System

We evaluate our localization system on six datasets covering 2,666 images, and in each dataset, the camera is moving at a distinct trajectory as UAV flies. The image is captured with a resolution of 1280×720 pixels at 10fps, using Intel RealSense (R200) camera looking ahead to perform building inspection, as shown in Fig. 7.

We adopt the UAV location in motion capture system

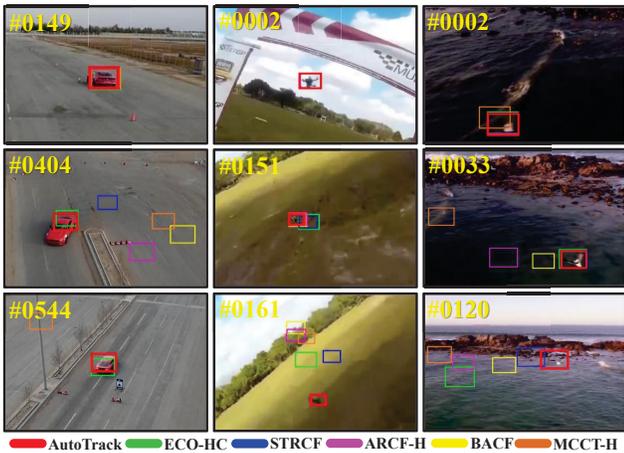


Figure 6. Screenshots of *Car16.2*, *ChasingDrones*, and *Gull1*.

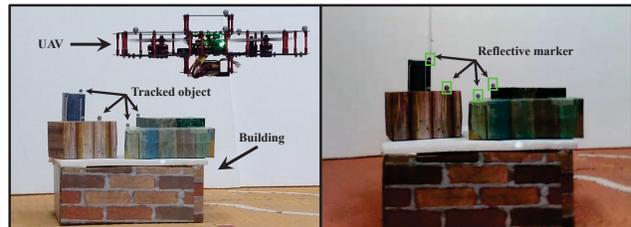


Figure 7. Experiment setup (left) and view from the UAV-mounted camera (right). The tracked objects (reflective markers whose ground truth locations are known in Quanser motion-capture system) for UAV localization are denoted as four green rectangles.

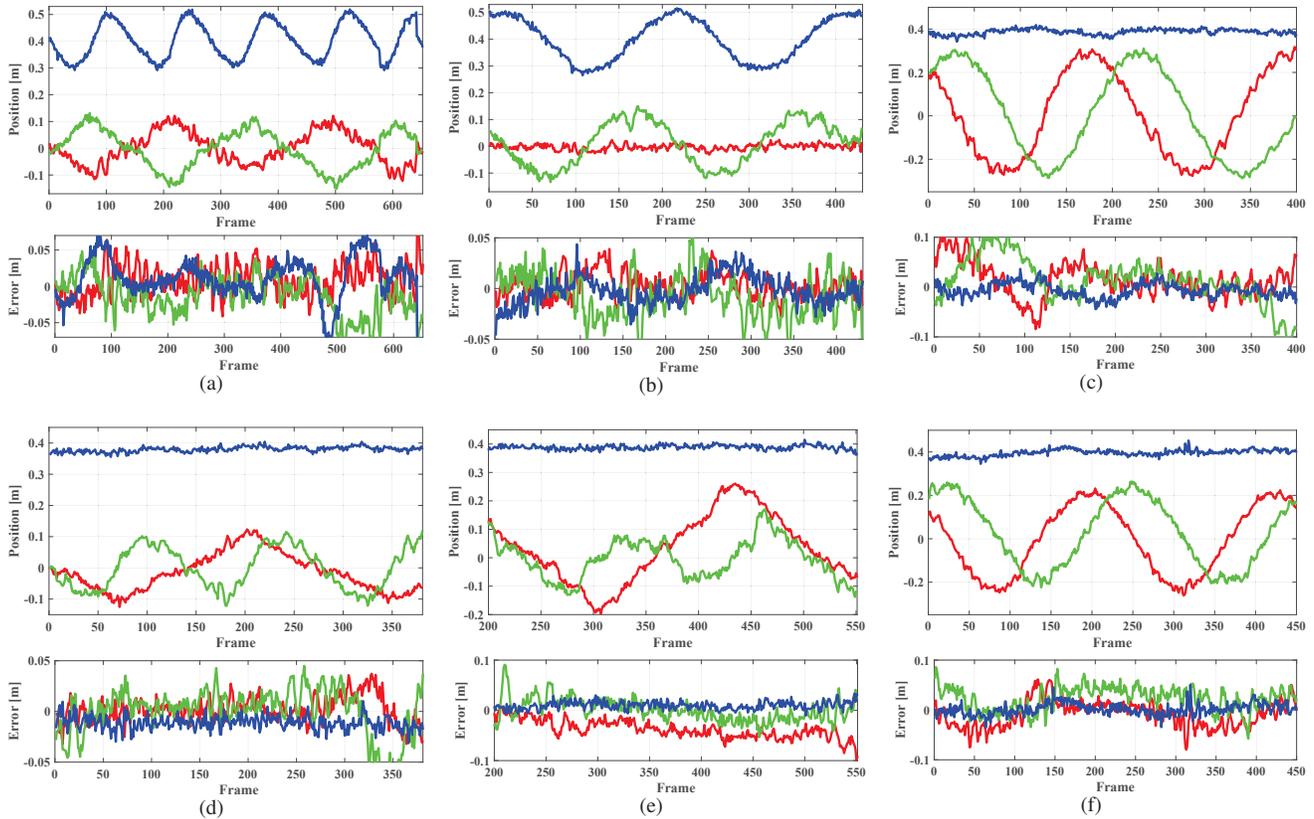


Figure 8. Estimation of camera position and the respective errors on six datasets. Lines with red, green and blue color denote x, y and z positions, respectively. The ground truth is not displayed because there is no noticeable differences with our results at such scale.

as the ground truth. The mean position errors in x, y and z directions are reported in Table 4. Figure 8 exhibits the estimated position as well as respective error in every frame. The root-mean-square error (RMSE) of our method on 2,666 frames is 3.44 centimeters.

Remark 4: It is noted that our system is applicable in various scenarios because our tracker can track any arbitrary objects once given their information in the first frame. In summary, compared to LED-based localization system [44], our method is more versatile and can run at real-time frame rates in the real-world scenarios.

Table 4. Illustration of estimation errors on six datasets covering 2,666 frames. The dataset is in line with the (a)-(f) in Fig. 8.

Dataset	x(cm)	y(cm)	z(cm)	RMSE	Frame number
(a)	1.90	2.25	2.38	3.79	652
(b)	1.06	1.88	1.13	2.44	431
(c)	3.01	3.51	1.30	4.80	400
(d)	1.01	1.73	1.16	2.32	381
(e)	3.77	1.77	1.05	4.30	352
(f)	2.27	2.77	0.91	3.69	450
Average	2.17	2.32	1.32	3.44	444

7. Conclusion

In this work, a generally applicable automatic spatio-temporal regularization framework is proposed for high-performance UAV tracking. Local response variation indicates local credibility, thus restricting local correlation filter learning. Global variation is able to control how much the correlation filter learns from the whole object. Comprehensive experiments have validated that AutoTrack is the best CPU-based tracker with a speed of ~ 60 fps, and even outperforms some state-of-the-art deep trackers on two UAV datasets [46, 47]. In addition, we try to bridge the gap between the theory and practice by utilizing visual tracking in UAV localization in the real world. Considerable tests proved the effectiveness and generality of our method. We strongly believe that our work can promote the development of visual tracking and its application in robotics.

Acknowledgment: This work is supported by the National Natural Science Foundation of China (No.61806148), the Fundamental Research Funds for the Central Universities (No.22120180009), and Tsinghua University Initiative Scientific Research Program.

References

- [1] Bonatti Rogerio, Ho Cherie, Wang Wenshan, Choudhury Sanjiban, and Scherer Sebastian. Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments. In *IROS*, pages 229–236, 2019. 1
- [2] Li Rui, Pang Minjian, Zhao Cong, Zhou Guyue, and Lu Fang. Monocular long-term target following on uavs. In *CVPR Workshops*, pages 29–37, 2016. 1
- [3] Changhong Fu, Adrian Carrio, Miguel A Olivares-Mendez, Ramon Suarez-Fernandez, and Pascual Campoy. Robust real-time vision-based aircraft tracking from unmanned aerial vehicles. In *ICRA*, pages 5441–5446, 2014. 1
- [4] Mücahit Karaduman, Ahmet Çınar, and Haluk Eren. Uav traffic patrolling via road detection and tracking in anonymous aerial video frames. *Journal of Intelligent & Robotic Systems*, 95(2):675–690, 2019. 1
- [5] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550, 2010. 1, 2
- [6] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, pages 254–265, 2014. 1, 2, 5
- [7] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Analysis and Machine Intelligence*, 37(3):583–96, 2015. 1, 2, 5
- [8] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318, 2015. 1, 2, 3, 5, 6
- [9] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, pages 1401–1409, 2016. 1
- [10] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *CVPR*, pages 1396–1404, 2017. 1, 5, 6
- [11] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1561–1575, 2017. 1, 5
- [12] Mengmeng Wang, Yong Liu, and Zeyi Huang. Large margin object tracking with circulant feature maps. In *CVPR*, pages 4021–4029, 2017. 1, 2
- [13] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, pages 4844–4853, 2018. 1, 5, 6
- [14] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, pages 4904–4913, 2018. 1, 2, 3, 5, 6
- [15] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016. 1, 2, 5
- [16] Sangdoon Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, pages 2711–2720, 2017. 1, 2, 5
- [17] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488, 2016. 1, 2, 5
- [18] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, pages 3074–3082, 2015. 1, 2, 5
- [19] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, pages 4335–4343, 2017. 1, 5
- [20] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017. 1, 2, 5, 6
- [21] Changhong Fu, Fuling Lin, Yiming Li, and Guang Chen. Correlation filter-based visual tracking for uav with online multi-feature learning. *Remote Sensing*, 11(5):549, 2019. 1
- [22] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, pages 1135–1143, 2017. 1, 2, 5, 6
- [23] Changhong Fu, Ziyuan Huang, Yiming Li, Ran Duan, and Peng Lu. Boundary effect-aware visual tracking for uav with online enhanced background learning and multi-frame consensus verification. In *IROS*, pages 4415–4422, 2019. 1, 2
- [24] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, pages 1430–1438, 2016. 1, 2, 5
- [25] Yiming Li, Changhong Fu, Ziyuan Huang, Yinqiang Zhang, and Jia Pan. Keyfilter-aware real-time uav object tracking. In *ICRA*, pages 1–7, 2020. 1, 2
- [26] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. In *ICCV*, pages 2891–2900, 2019. 1, 2, 3, 5, 6
- [27] Kaihua Zhang and Huihui Song. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*, 46(1):397–411, 2013. 2
- [28] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2016. 2
- [29] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715, 2012. 2

- [30] A. Lukežić, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, pages 4847–4856, 2017. 2, 5, 6
- [31] Kenan Dai, Dong Wang, Huchuan Lu, Chong Sun, and Jianhua Li. Visual tracking via adaptive spatially-regularized correlation filters. In *CVPR*, pages 4670–4679, 2019. 2, 5
- [32] Fan Li, Changhong Fu, Fuling Lin, Yiming Li, and Peng Lu. Training-set distillation for real-time uav object tracking. In *ICRA*, pages 1–7, 2020. 2
- [33] Yuxuan Sun, Chong Sun, Dong Wang, You He, and Huchuan Lu. Roi pooled correlation filters for visual tracking. In *CVPR*, pages 5783–5791, 2019. 2
- [34] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV Workshops*, pages 58–66, 2015. 2
- [35] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, pages 2555–2564, 2017. 2, 5
- [36] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, pages 8990–8999, 2018. 2
- [37] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 2
- [38] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, pages 4660–4669, 2019. 2
- [39] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, pages 4010–4019, 2019. 2
- [40] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6182–6191, 2019. 2
- [41] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, pages 4854–4863, 2018. 2
- [42] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, pages 105–114, 2017. 2
- [43] Andreas Breitenmoser, Laurent Kneip, and Roland Siegwart. A monocular vision-based system for 6d relative robot localization. In *IROS*, pages 79–85, 2011. 2
- [44] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared leds. In *ICRA*, pages 907–913, 2014. 2, 4, 8
- [45] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, 3:1–122, 2011. 3
- [46] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, pages 1–7, 2017. 5, 6, 8
- [47] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: object detection and tracking. In *ECCV*, pages 370–386, 2018. 5, 6, 8
- [48] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461, 2016. 5, 6
- [49] Longyin Wen, Pengfei Zhu, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Chenfeng Liu, Hao Cheng, Xiaoyu Liu, Wenya Ma, et al. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *ECCV Workshops*, pages 1–27, 2018. 5, 6
- [50] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, pages 1–6, 2009. 5
- [51] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, and Ming-Hsuan Yang. Target-aware deep tracking. In *CVPR*, pages 1369–1378, 2019. 5
- [52] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 2805–2813, 2017. 5
- [53] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, pages 1308–1317, 2019. 5
- [54] Feng Li, Yingjie Yao, Peihua Li, David Zhang, Wangmeng Zuo, and Ming-Hsuan Yang. Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In *ICCV Workshops*, pages 2001–2009, 2017. 5
- [55] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016. 5
- [56] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, 2017. 5
- [57] Chen Wang, Le Zhang, Lihua Xie, and Junsong Yuan. Kernel cross-correlator. In *AAAI*, pages 4179–4186, 2018. 5
- [58] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, pages 1–11, 2014. 5
- [59] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418, 2013. 6