2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Keyfilter-Aware Real-Time UAV Object Tracking

Yiming Li[1], Changhong Fu[1,*], Ziyuan Huang[2], Yinqiang Zhang[3], and Jia Pan[4]

***Abstract*— Correlation filter-based tracking has been widely applied in unmanned aerial vehicle (UAV) with high efficiency. However, it has two imperfections, i.e., boundary effect and filter corruption. Several methods enlarging the search area can mitigate boundary effect, yet introducing undesired background distraction. Existing frame-by-frame context learning strategies for repressing background distraction nevertheless lower the tracking speed. Inspired by keyframe-based simultaneous localization and mapping, keyfilter is proposed in visual tracking for the first time, in order to handle the above issues efficiently and effectively. Keyfilters generated by periodically selected keyframes learn the context intermittently and are used to restrain the learning of filters, so that 1) context awareness can be transmitted to all the filters via keyfilter restriction, and 2) filter corruption can be repressed. Compared to the state-of-the-art results, our tracker performs better on two challenging benchmarks, with enough speed for UAV real-time applications.**

I. INTRODUCTION

Combined with extensibility, autonomy, and maneuverability of unmanned aerial vehicle (UAV), visual object tracking has considerable applications in UAV, e.g., person tracing [1], autonomous landing [2], aerial photography [3], and aircraft tracking [4]. Notwithstanding some progress, UAV tracking remains onerous because of the complex background, frequent appearance variation caused by UAV motion, full/partial occlusion, deformation, as well as illumination changes. Besides, computationally intractable trackers are not deployable onboard UAVs because of the harsh calculation resources and limited power capacity.

Recently, the framework of discriminative correlation filter (DCF) [5], aiming to discriminate the foreground from the background via a correlation filter (CF), is widely adopted in UAV object tracking. The speed is hugely raised because of its utilization of the circulant matrices' property to carry out the otherwise cumbersome calculation in the frequency domain rather than spatial one. Yet the circulant artificial samples used to train the filter hamper the filter's discriminative ability. This problem is called boundary effect because the artificial non-real samples have periodical splicing at the boundary. Several approaches [6]–[13] expand the search area for alleviating boundary effects, but the enlargement has introduced more context noise, distracting the detection phase especially in situations of similar objects around.

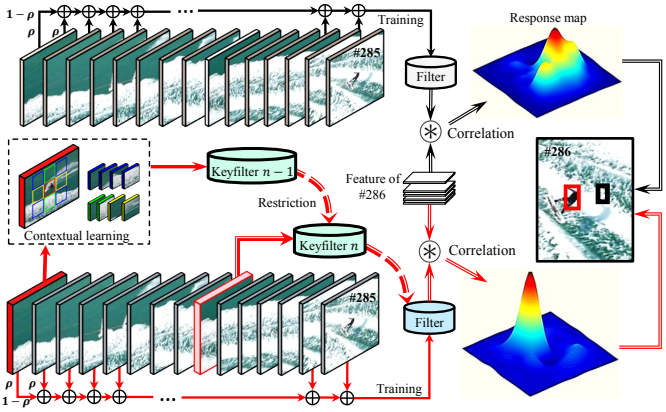

Fig. 1. Comparison between response maps of our tracker and baseline. Red frames are served as keyframes generating keyfilters. Keyfilters carry out context learning intermittently and influence the current filter training for mitigating filter corruption. Feature of current frame is correlated with the filter trained in the last frame, producing a response map. Red and black rectangles denote respectively the results from KAOT and baseline.

In literature, the context-aware framework [14] is proposed to reduce the context distraction through response repression of the context patches. However, the frame-by-frame context learning is extremely redundant, because the capture frequency of drone camera is generally smaller than the frequency of context variation, e.g., the interval time between two consecutive time in a 30 frame per second (FPS) video is 0.03 second, but generally the context appearance in aerial view remains unchanged for a certain time far more than 0.03 second. In addition, the learned single filter without restriction is prone to corruption due to the omnipresent appearance variations in the aerial scenarios.

In this work, inspired by keyframe-based simultaneous localization and mapping (SLAM) [15], the keyframe technique is used to raise the tracking performance efficiently and effectively. The contributions of this work are two-fold:

- A novel application of the keyfilter in UAV visual object tracking is presented. Keyfilters generated at a certain frequency learn the context intermittently and enforce temporal restriction. Through the restriction, the filter corruption in the time span is alleviated and context noise is efficiently suppressed.
- Extensive experiments on 193 challenging UAV image sequences have shown that the **k**eyfilter-**a**ware **o**bject **t**racker, i.e., KAOT, has competent performance compared with the state-of-the-art tracking approaches based on DCF and deep neural network (DNN).

[1]Yiming Li and Changhong Fu are with the School of Mechanical Engineering, Tongji University, 201804 Shanghai, China. changhongfu@tongji.edu.cn

[2]Ziyuan Huang is with the Advanced Robotics Centre, National University of Singapore, Singapore. ziyuan.huang@u.nus.edu

[3]Yinqiang Zhang is with the Department of Mechanical Engineering, Technical University of Munich, Munich, Germany. yinqiang.zhang@tum.de

[4]Jia Pan is with the Computer Science Department, The University of Hong Kong, Hong Kong, China. panjia1983@gmail.com

II. RELATED WORKS

A. Discriminative correlation filter

In recent years, the framework of discriminative correlation filter (DCF) [5] has broadly aroused research interest due to its remarkable efficiency. Yet classic CF-based trackers [16]–[18] have limited performance due to the lack of negative samples, i.e., the circulant artificial samples created to train the CF hugely reduce its discriminative power. One solution to this problem is spatial penalization to punish the filter value at the boundary [6]–[10]. Another solution is cropping both the background and target to use negative samples in the real word instead of synthetic samples [11]–[13]. However, the aforementioned approaches are prone to introduce context distraction because of enlarging search area, especially in the scenarios of similar object around.

B. Prior work to context noise and filter corruption

In literature, M. Mueller et al. [14] proposed to repress the response of context patches, i.e., the features extracted from surrounding context are directly fed into classic DCF framework and their desired responses are suppressed as zero. The context distraction is thus effectively repressed, consequently the discriminative ability of the filter is enhanced. Nevertheless, the frame-by-frame context learning is effective but not efficient, and its redundancy can be significantly reduced. Another problem of classic DCF trackers is that the learned single filter is commonly subjected to corruption because of the frequent appearance variation. Online passive-aggressive learning is incorporated into the DCF framework [19] to mitigate the corruption. Compared to [19], the presented keyfilter performs better in both precision and speed.

C. Tracking by deep neural network

Recently, deep neural network has contributed a lot to the development of computer vision. For visual tracking, some deep trackers [20]–[22] fine-tuning the deep network online for high precision yet run too slow (around 1 fps on a high-end GPU) to use in practice. Other methods like deep reinforcement learning [23], unsupervised learning [24], continues operator [8], end-to-end learning [25] and deep feature representation [26] have also increased the tracking accuracy. Among them, incorporating lightweight deep features into online learned DCF framework has exhibited competitive performance both in precision and efficiency.

D. Tracking for unmanned aerial vehicle

Mechanical vibration, motion blur, limited computation capacity and rapid movement have made UAV tracking an extremely demanding task. In literature, the presented UAV-tailored tracking methods generally have lower robustness and accuracy [4], [27]–[29]. In light of offline training on the large-scale image datasets, deep feature for robust representation can improve performance significantly, yet the speed of existing deep-feature based trackers mostly run slow even on a high-end GPU [9]. This work aims to improve the speed and accuracy for the deep feature-based DCF framework for real-time UAV applications.

III. REVIEW OF BACKGROUND-AWARE CORRELATION FILTER

The objective function of background-aware correlation filters (BACF) [12] is as follows:

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{d=1}^{D} \mathbf{B}\mathbf{x}_0^d \star \mathbf{w}^d\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{D}\|\mathbf{w}^d\|_2^2 \,, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{x}^d \in \mathbb{R}^N$ and $\mathbf{w}^d \in \mathbb{R}^M$ denote the desired response, the dth one of D feature channels and correlation filter respectively. λ is a regularization parameter and $\mathcal{E}(\mathbf{w})$ refers to an error between the desired response $\mathbf{y}$ and the actual one. $\star$ is the spatial correlation operator. The main idea of BACF is to utilize a cropping matrix $\mathbf{B} \in \mathbb{R}^{M\times N}$ to extract real negative samples. However, more background distraction is introduced because of the enlargement.

IV. KEYFILTER-AWARE OBJECT TRACKER

Inspired by the keyframe technique used in SLAM, the keyfilter is firstly proposed in visual tracking to boost accuracy and efficiency, as illustrated in Fig. 2. The objective function of KAOT tracker is written as follows:

$$\begin{aligned}\mathcal{E}(\mathbf{w}) = &\frac{1}{2}\|\mathbf{y} - \sum_{d=1}^{D} \mathbf{B}\mathbf{x}_0^d \star \mathbf{w}^d\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{D}\|\mathbf{w}^d\|_2^2 \\ &+ \frac{S_p}{2}\sum_{p=1}^{P}\|\sum_{d=1}^{D}\mathbf{B}\mathbf{x}_p^d \star \mathbf{w}^d\|_2^2 + \frac{\gamma}{2}\sum_{d=1}^{D}\|\mathbf{w}^d - \tilde{\mathbf{w}}^d\|_2^2\end{aligned} \,, \tag{2}$$

where the third term is response repression of context patches (their desired responses are zero), and S_p is the score of pth patch to measure the necessity of penalization (introduced in IV-B). $\mathbf{w}^d \in \mathbb{R}^M$ and $\tilde{\mathbf{w}}^d \in \mathbb{R}^M$ are the current filter and keyfilter, respectively. γ is the penalty parameter of the gap between $\mathbf{w}^d$ and $\tilde{\mathbf{w}}^d$. To improve the calculation speed, Eq. (2) is calculated in the frequency domain:

$$\begin{aligned}\mathcal{E}(\mathbf{w}, \hat{\mathbf{g}}) = &\frac{1}{2}\|\hat{\mathbf{X}}\hat{\mathbf{g}} - \hat{\mathbf{Y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w} - \tilde{\mathbf{w}}\|_2^2 \\ &s.t. \quad \hat{\mathbf{g}} = \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\end{aligned} \,, \tag{3}$$

where $\otimes$ is the Kronecker product and $\mathbf{I}_D \in \mathbb{R}^{D\times D}$ is an identity matrix. $\hat{}$ denotes the discrete Fourier transform with orthogonal matrix $\mathbf{F}$. $\hat{\mathbf{X}}^T = \left[\, \hat{\mathbf{X}}_0, \hat{S_1\mathbf{X}}_1, \cdots, \hat{S_p\mathbf{X}}_P \,\right]$, $\hat{\mathbf{Y}} = \left[\, \hat{\mathbf{y}}, \mathbf{0}, \cdots, \mathbf{0} \,\right]$, and $\hat{\mathbf{X}}_p \in \mathbb{C}^{N\times DN}(p = 0, 1, ..., P)$, $\hat{\mathbf{g}} \in \mathbb{C}^{DN\times 1}$ and $\mathbf{w} \in \mathbb{R}^{DM\times 1}$ are respectively defined as $\hat{\mathbf{X}} = [diag(\hat{\mathbf{x}}^1)^\top, \cdots, diag(\hat{\mathbf{x}}^D)^\top]$, $\hat{\mathbf{g}} = [\hat{\mathbf{g}}^{1\top}, \cdots, \hat{\mathbf{g}}^{D\top}]^\top$, $\tilde{\mathbf{w}} = [\tilde{\mathbf{w}}^{1\top}, \cdots, \tilde{\mathbf{w}}^{D\top}]^\top$ and $\mathbf{w} = [\mathbf{w}^{1\top}, \cdots, \mathbf{w}^{D\top}]^\top$.

A. Optimization algorithm

Equation (3) can be optimized via alternating direction method of multipliers (ADMM) [30]. The Augmented Lagrangian form of Eq. (3) is:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \hat{\mathbf{g}}, \hat{\boldsymbol{\zeta}}) = &\frac{1}{2}\|\hat{\mathbf{X}}\hat{\mathbf{g}} - \hat{\mathbf{Y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w} - \tilde{\mathbf{w}}\|_2^2 \\ &+ \hat{\boldsymbol{\zeta}}^\top\left(\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\right) \\ &+ \frac{\mu}{2}\|\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\|_2^2\end{aligned} \,, \tag{4}$$

where $\hat{\boldsymbol{\zeta}} \in \mathbb{C}^{DN\times 1}$ is the Lagrangian vector in the frequency domain and μ is a penalty parameter. Two subproblems $\hat{\mathbf{g}}^*$ and $\mathbf{w}^*$ are solved alternatively.

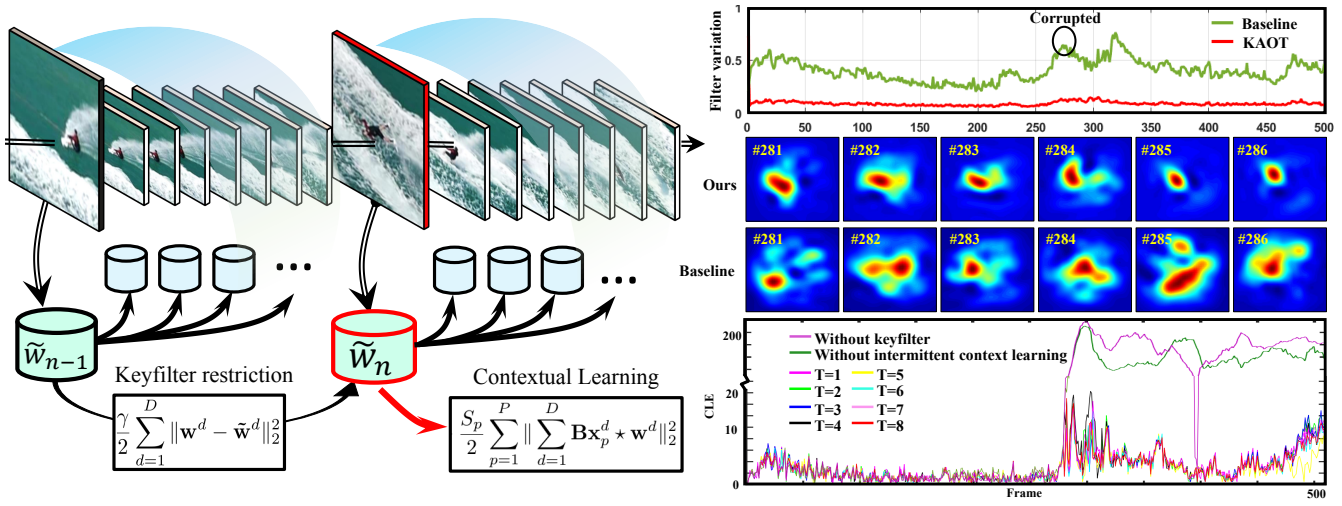

Fig. 2. **Illustration of advantages of KAOT.** With the keyfilter restriction, the filter corruption is mitigated, as shown on the top right. With the context learning, the distraction is reduced, as shown in the response maps from frame 281 to 286. Set the keyfilter update period T as 1-8 frames (learns the context every 2 - 16 frames), and the object is tracked successfully in all eight trackers, while FPS (frame per second) is raised to 15.2 from 9.8, lowering the redundancy of context learning significantly. In addition, trackers lacking of the keyfilter restriction or the context learning both lose the target.

• **Subproblem $\mathbf{w}^*$ (filter in the spatial domain):**

$$\begin{aligned}\mathbf{w}^* = \arg\min_{\mathbf{w}} \Big\{ & \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w}-\tilde{\mathbf{w}}\|_2^2 \\ & + \hat{\boldsymbol{\zeta}}^\top\big(\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\big) \\ & + \frac{\mu}{2}\|\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\|_2^2 \Big\} \\ = & \left(\mu + \frac{\lambda+\gamma}{N}\right)^{-1}(\mu\mathbf{g} + \boldsymbol{\zeta} + \frac{\gamma}{N}\tilde{\mathbf{w}})\end{aligned} \tag{5}$$

• **Subproblem $\hat{\mathbf{g}}^*$ (filter in the frequency domain):**

$$\begin{aligned}\hat{\mathbf{g}}^* = \arg\min_{\hat{\mathbf{g}}} \Big\{ & \frac{1}{2}\|\hat{\mathbf{X}}\hat{\mathbf{g}} - \hat{\mathbf{Y}}\|_2^2 \\ & + \hat{\boldsymbol{\zeta}}^\top\big(\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\big) \\ & + \frac{\mu}{2}\|\hat{\mathbf{g}} - \sqrt{N}(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top)\mathbf{w}\|_2^2 \Big\}\end{aligned} \tag{6}$$

$\hat{\mathbf{y}}(n)$ only depends on $\hat{\mathbf{x}}(n) = \left[\hat{\mathbf{x}}^1(n), \hat{\mathbf{x}}^2(n), ..., \hat{\mathbf{x}}^D(n)\right]^\top$ and $\hat{\mathbf{g}}(n) = \left[\text{conj}\left(\hat{\mathbf{g}}^1(n)\right), \ldots, \text{conj}\left(\hat{\mathbf{g}}^D(n)\right)\right]^\top$. Hence, solving equation for $\hat{\mathbf{g}}^*$ can be identically written as N separate functions $\hat{\mathbf{g}}(n)$ $(n = [1, ..., N])$:

$$\begin{aligned}\hat{\mathbf{g}}(n)^* = \arg\min_{\hat{\mathbf{g}}(n)} \Big\{ & \frac{1}{2}\|\hat{\mathbf{y}}(n) - \hat{\mathbf{x}}_0(n)^\top\hat{\mathbf{g}}(n)\|_2^2 \\ & + \frac{1}{2}\sum_{p=1}^{P}\|S_p\hat{\mathbf{x}}_p(n)^\top\hat{\mathbf{g}}(n)\|_2^2 + \hat{\zeta}(n)^\top(\hat{\mathbf{g}}(n) - \hat{\mathbf{w}}(n)) \\ & + \frac{\mu}{2}\|\hat{\mathbf{g}}(n) - \hat{\mathbf{w}}(n)\|_2^2 \Big\}\end{aligned} \tag{7}$$

where $\hat{\mathbf{w}}(n) = \left[\hat{\mathbf{w}}^1(n), \ldots, \hat{\mathbf{w}}^D(n)\right]$ and $\hat{\mathbf{w}}^d = \sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{w}^d$. The solution to each sub-subproblem is:

$$\begin{aligned}\hat{\mathbf{g}}(n)^* = & (\sum_{p=0}^{P} S_p^2\hat{\mathbf{x}}_p(n)\hat{\mathbf{x}}_p(n)^\top + \mu\mathbf{I}_D)^{-1} \\ & (\hat{\mathbf{y}}(n)\hat{\mathbf{x}}_0(n) - \hat{\zeta}(n) + \mu\hat{\mathbf{w}}(n))\end{aligned} \tag{8}$$

Lagrangian parameter is updated as follows:

$$\hat{\zeta}_{j+1} = \hat{\zeta}_j + \mu\left(\hat{\mathbf{g}}^*_{j+1} - \hat{\mathbf{w}}^*_{j+1}\right), \tag{9}$$

and $\hat{\mathbf{w}}^*_{j+1}$ is obtained through the following formula:

$$\hat{\mathbf{w}}^*_{j+1} = \left(\mathbf{I}_D \otimes \mathbf{F}\mathbf{B}^\top\right)\mathbf{w}^*_{j+1}, \tag{10}$$

subscript j denotes the the value at last iteration and subscript $j+1$ denotes the value at current iteration.

B. Context patches scoring scheme

This work adopts a simple but effective scheme for measuring the score of context patches through Euclidean distance. Specifically, the size of omni-directional patches located around the object is the same as that of the object. The score of patch p is calculated as follows:

$$S_p = \frac{min\{w, h\}}{|OO_p|}s\ , \tag{11}$$

where $|OO_p|$ denotes the Euclidean distance between the object and context patch p $(p = 1, 2, ..., P)$ (between center points) and s is the base score which is a constant number. w, h are respectively the width and height of the object rectangle. Through Eq. (11) , the patch which is closer to object, obtains a higher score for stronger penalization.

C. Keyfilter updating strategy

Starting from the first frame, the keyfilter is generated at a certain frequency using keyframes and current keyfilter refers to the latest trained keyfilter, as shown in Fig. 2. Current filter is restricted by current keyfilter through the punishment introduced by the gap between current filter and keyfilter. In other words, current keyfilter is updated every c frames ($c = 8$ in this work). When the $(n+1)$th keyframe arrives (frame $k = c \times n + 1$), the filter of current frame (keyfilter $(n+1)$) is trained under influence from the keyfilter n. As for the non-keyframes after keyfilter $(n+1)$, the filters of them are learned with the restriction of current keyfilter (keyfilter $(n+1)$). The detailed work-flow of KAOT tracker is presented in Algorithm 1.

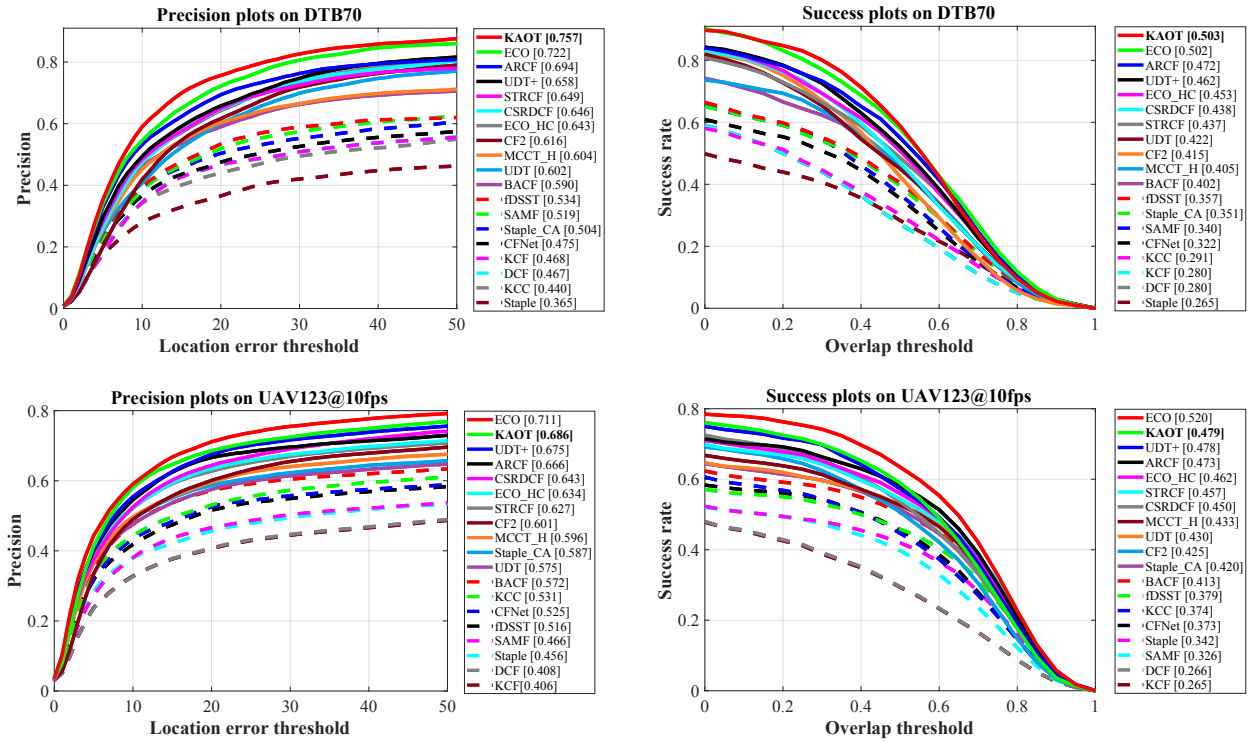

Fig. 3. **Precision and success plots** based on one-pass-evaluation [31] of KAOT and other real-time trackers on DTB70 [32] and UAV123@10fps [33].

Algorithm 1: KAOT tracker

Input: Location of tracked object on frame $k-1$,
Current keyfilter $\tilde{\mathbf{w}}$,
Keyfilter updating $Stepsize$.

Output: Location and scale of object on frame k

for $i = 2$ *to end* **do**
  Extract features from the region of interest (ROI)
  Convolute $\hat{\mathbf{g}}_{k-1}$ with $\hat{\mathbf{x}}^i_{\text{detect}}$ on different scales to generate response maps
  Find the peak position of map and output
  Update object model
  if k *mod Stepsize*$\times 2 == 0$ **then**
    Calculate S_p $(p = 1, 2, ..., 8)$ by Eq. (11)
    Learn CF $\mathbf{w}_k$ by Eq. (5), Eq. (8) and Eq. (9)
    $\tilde{\mathbf{w}} = \mathbf{w}_k$
  else
    if k *mod Stepsize* $== 0$ **then**
      $S_p = 0$ $(p = 1, 2, ..., 8)$
      Learn $\mathbf{w}_k$ by Eq. (5), (8) and Eq. (9)
      $\tilde{\mathbf{w}} = \mathbf{w}_k$
    else
      $S_p = 0$ $(p = 1, 2, ..., 8)$
      Learn $\mathbf{w}_k$ by Eq. (5), Eq. (8) and Eq. (9)
    end
  end
  Start detection of next frame
end

V. EXPERIMENTS

In this section, the presented KAOT tracker is rigorously evaluated on two difficult UAV datasets, i.e., DTB70 [32] and UAV123@10ps [33], with overall 193 image sequences captured by drone camera. The tracking results are compared with the state-of-the-art trackers including both real-time (>=12 FPS) and non-real-time (< 12FPS) ones, i.e., ARCF [13], UDT [24], UDT+ [24], MCCT [34], MCCT-H [34], CSR-DCF [10], STRCF [19], DeepSRTCF [19], ECO [8], ECO-HC [8], BACF [12], Staple [16], Staple-CA [14], CF2 [26], DCF [14], DSST [35], KCF [5], KCC [36], SAMF [17], ADNet [23], CFNet [25], MCPF [37], IBCCF [38]. This work evaluates the trackers based on protocol in two datasets respectively [32], [33]. Noted that the real-time trackers are trackers with enough speed for UAV real-time applications.

A. Implementation details

KAOT adopts both the hand-crafted and deep features, i.e., histogram oriented gradient (HOG) [39], color name (CN) [40] and conv3 layer from VGG-M network [41]. The value of γ is set as 10, and the base score s is set as 0.28. ADMM iteration is set to 2 for raising efficiency. All trackers are implemented in MATLAB R2018a and all the experiments are conducted on the same computer with an i7-8700K processor (3.7GHz), 48GB RAM and NVIDIA GTX 2080 GPU. It is noted that the original codes without any modification are employed in this work for fair comparison.

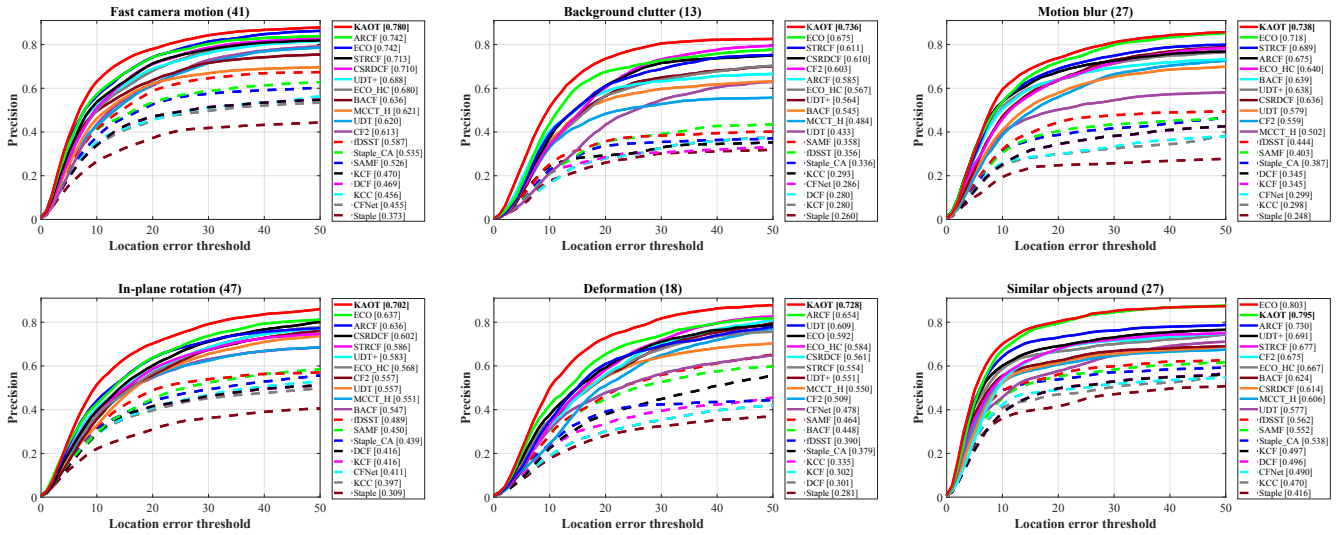

Fig. 4. **Attribute based evaluation on precision.** KAOT ranks first place on five out of six challenging attributes.

TABLE I

AVERAGE PRECISION (THRESHOLD AT 20 PIXELS) AND SPEED ((FPS, * MEANS GPU SPEED, OTHERWISE CPU SPEED)) OF TOP TEN REAL-TIME TRACKERS. RED , GREEN, AND BLUE FONTS RESPECTIVELY INDICATES THE BEST, SECOND, AND THIRD PLACE IN TEN TRACKERS.

	KAOT	ECO [8]	ARCF [13]	UDT+ [24]	STRCF [19]	CSRDCF [10]	ECO_HC [8]	CF2 [26]	MCCT_H [34]	UDT [24]
Avg. precision	72.2	71.7	68.0	66.5	63.8	63.5	64.3	62.5	60.3	58.9
Speed (FPS)	14.7*	11.6*	15.3	43.4*	26.3	11.8	62.19	14.4*	59.0	57.5*

B. Comparison with real-time trackers

1) Overall performance: Figure 3 demonstrates the overall performance of KAOT with other state-of-the-art real-time trackers on DTB70 and UAV123@10fps. On DTB70 dataset, KAOT (0.757) has an advantage of 4.4% and 9.1% over the second and third best tracker ECO (0.722), ARCF (0.694) respectively in precision, along with a gain of 0.2% and 6.6% over the second (ECO, 0.502) and third best tracker (ARCF, 0.472) respectively in AUC. On UAV123@10fps dataset, KAOT (0.686, 0.479) ranks second place followed by the third place UDT+ (0.675, 0.478). ECO is the only tracker performing better than KAOT. Nevertheless, it utilizes continuous operator to fuse the feature maps elaborately, while KAOT just uses the simple BACF as baseline. Notice that ECO can further enhance its performance with our framework. Average precision on the two datasets and speed (evaluated on DTB70) are reported in Table I. KAOT is 27% faster than ECO when achieving higher precision.

Discussions: DTB70 [32] dataset is recorded on a drone with more frequent and drastic displacements compared to UAV123@10fps [33], thus increasing the tracking difficulties. Our method exhibits relatively big advantages on DTB70, proving the robustness of our method in the scenarios of strong motion.

2) Attribute-based performance: Precision plots of six challenging attributes are demonstrated in Figure 4. In the cases of background clutter, KAOT improves the ECO by 9.0% in light of the intermittent context learning which can suppress the background distraction effectively. In situations of in-plane rotation and deformation, KAOT has a superiority

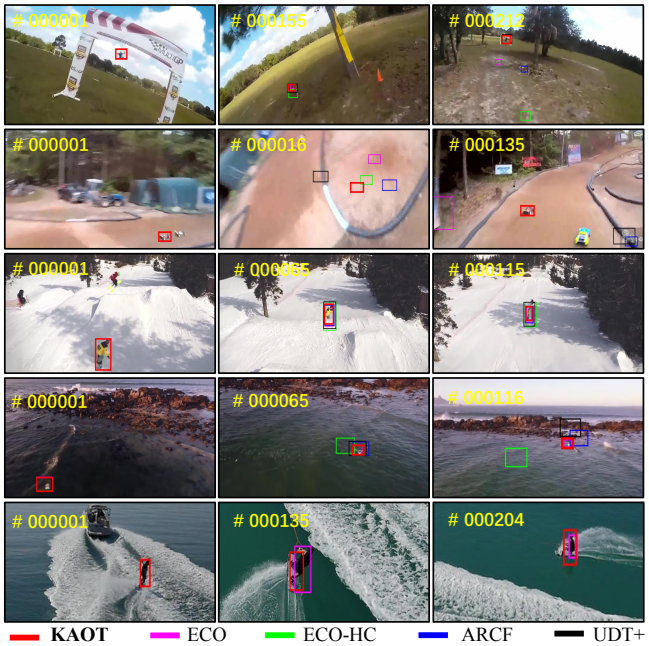

Fig. 5. **Qualitative evaluation.** From the top to bottom is respectively the sequence *ChasingDrones*, *RcCar6*, *SnowBoarding2*, *Gull1* and *wakeboard2*. Code and UAV tracking video are: https://github.com/vision4robotics/KAOT-tracker and https://youtu.be/jMfmHVRqv3Y.

of 10.2% and 23.0% respectively compared to ECO. This is attributed to the keyfilter restriction, which can prevent the filter from aberrant variation. In addition, KAOT exhibits excellent performance in the scenario of fast camera motion and motion blur, which is desirable in aerial tracking.

3) Qualitative evaluation: Qualitative tracking results on five difficult UAV image sequences are shown in Figure 5. Besides, the respective center location error (CLE) variations of five sequences are visualized in Figure 6. Specifically, in $ChasingDrones$ sequence where tracking is bothered by strong UAV motion, KAOT has effectively repressed the distraction of the context, so it can perform well despite the large movement in a certain complex context. Only the pre-trained UDT+ tracks successfully in addition to KAOT. Motion blur occurs in sequences $RcCar6$ and $Gull1$ (severe example is shown at frame 16 in $RcCar6$). In this situation, KAOT has kept tracking owing to the mitigated filter corruption. As for the last two sequences, keyfilter restriction and intermittent context learning have collaboratively contributed to successful tracking.

C. Comparison with non-real-time trackers

KAOT is also compared with five non-real-time trackers using deep neural network, as shown in Table II. To sum up, KAOT has the best performance in terms of both precision and speed on two benchmarks. In addition, compared to DeepSTRCF (using the same features as KAOT), our tracker has more robust performance in precision on both two datasets and is around 2.4 times faster than it. Therefore, the efficiency and accuracy of KAOT tracker can be proven.

D. Limitations and future works

Keyframe selection: This work only adopts a simple periodic keyframe selection mechanism, which is possible to introduce distraction when the tracking on the keyframes is not reliable. More elaborated strategy can be employed to adaptively choose the keyframe and further enhance the robustness.

Re-detection and rotation: Though KAOT performs favorably in the situations of drastic appearance change like blur, deformation, etc., it is still limited when the object disappear for a long time. Also, KAOT can not handle the rotation situations. Thus the re-detection and rotation-aware modules can be added to raise the performance.

Speed: The speed of KAOT is around 15 fps with a GPU and can be used in real-time applications. However, KAOT tracker is implemented on MATLAB platform and the code is not optimized, so the speed can be further improved.

TABLE II

PRECISION, SUCCESS RATE (THE AREA UNDER THE CURVE), AND FPS OF KAOT AS WELL AS FIVE NON-REAL-TIME TRACKERS. RED , GREEN, AND BLUE FONTS RESPECTIVELY INDICATES THE BEST, SECOND, AND THIRD PERFORMANCE.

	DTB70		UAV123@10fps		FPS
Trackers	Prec.	AUC	Prec.	AUC	
MCPF [37]	66.4	43.3	66.5	44.5	0.57*
MCCT [34]	72.5	48.4	68.4	49.2	8.49*
DeepSTRCF [19]	73.4	50.6	68.2	49.9	6.18*
IBCCF [38]	66.9	46.0	65.1	48.1	2.28*
ADNet [23]	63.7	42.2	62.5	43.9	6.87*
KAOT	75.7	50.3	68.6	47.9	14.69*

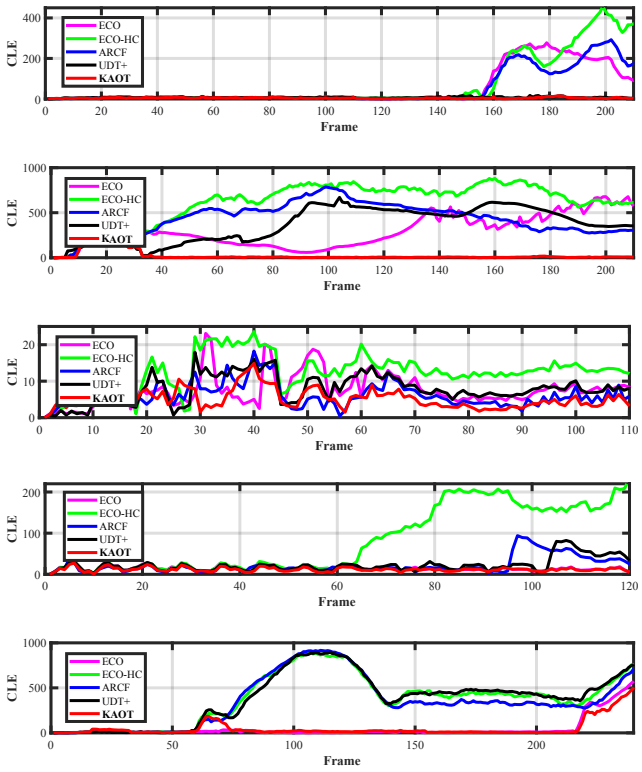

Fig. 6. **Illustration of CLE variations.** From top to bottom is the result from sequence $ChasingDrones$, $RcCar6$, $SnowBoarding2$, $Gull1$ and $wakeboard2$, respectively.

VI. CONCLUSIONS

This work proposes keyfilter-aware object tracker to repress the filter corruption and lower the redundancy of context learning. Extensive experiments on two authoritative datasets have validated our tracker performs favorably in precision, with enough speed for real-time applications. This keyfilter-aware framework and intermittent context learning strategy can also be used in other trackers like C-COT [7] and STRCF [19] to further boost their performance. We strongly believe that our method can be used in practice and promote the development of UAV tracking applications.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61806148) and the Fundamental Research Funds for the Central Universities (No. 22120180009).

REFERENCES

[1] H. Cheng, L. Lin, Z. Zheng, Y. Guan, and Z. Liu, "An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1732–1738.

[2] C. Fu, A. Carrio, M. A. Olivares-Mendez, and P. Campoy, "Online learning-based robust visual tracking for autonomous landing of Unmanned Aerial Vehicles," in *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 649–655.

[3] M. Gschwindt, E. Camci, R. Bonatti, W. Wang, E. Kayacan, and S. Scherer, "Can a Robot Become a Movie Director? Learning Artistic Principles for Aerial Cinematography," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[4] C. Fu, A. Carrio, M. A. Olivares-Méndez, R. Suarez-Fernandez, and P. C. Cervera, "Robust Real-time Vision-based Aircraft Tracking From Unmanned Aerial Vehicles," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5441–5446.

[5] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 583–596, 2015.

[6] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning Spatially Regularized Correlation Filters for Visual Tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4310–4318.

[7] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking," in *European Conference on Computer Vision*, 2016, pp. 472–488.

[8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6931–6939.

[9] C. Fu, Z. Huang, Y. Li, R. Duan, and P. Lu, "Boundary Effect-Aware Visual Tracking for UAV with Online Enhanced Background Learning and Multi-Frame Consensus Verification," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[10] A. Lukežic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4847–4856.

[11] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Correlation Filters With Limited Boundaries," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[12] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning Background-Aware Correlation Filters for Visual Tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1144–1152.

[13] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning Aberrance Repressed Correlation Filters for Real-Time UAV Tracking," in *2019 IEEE International Conference on Computer Vision (ICCV)*, 2019.

[14] M. Mueller, N. Smith, and B. Ghanem, "Context-Aware Correlation Filter Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1387–1395.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, 2015.

[16] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary Learners for Real-Time Tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1401–1409.

[17] Y. Li and J. Zhu, "A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration," in *Proceedings of European Conference on Computer Vision Workshops*, 2015, pp. 254–265.

[18] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-Term Correlation Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5388–5396.

[19] F. Li, C. Tian, W. Zuo, L. Zhang, and M. Yang, "Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913.

[20] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[21] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4293–4302.

[22] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3119–3127.

[23] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1349–1358.

[24] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised Deep Tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2805–2813.

[26] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical Convolutional Features for Visual Tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.

[27] Y. Yin, X. Wang, D. Xu, F. Liu, Y. Wang, and W. Wu, "Robust Visual Detection-Learning-Tracking Framework for Autonomous Aerial Refueling of UAVs," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, pp. 510–521, 2016.

[28] C. Yuan, Z. Liu, and Y. Zhang, "UAV-based forest fire detection and tracking using image processing techniques," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 639–643.

[29] C. Martinez, I. F. Mondragon, P. C. Cervera, J. L. Sanchez-Lopez, and M. A. Olivares-Mendez, "A Hierarchical Tracking Strategy for Vision-Based Applications On-Board UAVs," *Journal of Intelligent and Robotic Systems*, vol. 72, pp. 517–539, 2013.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," in *Foundations and Trends in Machine Learning*, vol. 3, 2010, pp. 1–122.

[31] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1834–1848, 2015.

[32] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *AAAI*, 2017.

[33] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2016, pp. 445–461.

[34] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue Correlation Filters for Robust Visual Tracking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4844–4853.

[35] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative Scale Space Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[36] C. Wang, L. Zhang, L. Xie, and J. Yuan, "Kernel cross-correlator," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[37] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4335–4343.

[38] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M. Yang, "Integrating Boundary and Center Correlation Filters for Visual Tracking with Aspect Ratio Variation," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 2001–2009.

[39] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893.

[40] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive Color Attributes for Real-Time Visual Tracking," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, 2014.

[41] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of International Conference on Representation Learning*, 2015, pp. 1–14.