Training-Set Distillation for Real-Time UAV Object Tracking

Fan Li¹, Changhong Fu¹,* Fuling Lin¹, Yiming Li¹, and Peng Lu²

Abstract—Correlation filter (CF) has recently exhibited promising performance in visual object tracking for unmanned aerial vehicle (UAV). Such online learning method heavily depends on the quality of the training-set, yet complicated aerial scenarios like occlusion or out of view can reduce its reliability. In this work, a novel time slot-based distillation approach is proposed to efficiently and effectively optimize the training-set’s quality on the fly. A cooperative energy minimization function is established to score the historical samples adaptively. To accelerate the scoring process, frames with high confident tracking results are employed as the keyframes to divide the tracking process into multiple time slots. After the establishment of a new slot, the weighted fusion of the previous samples generates one key-sample, in order to reduce the number of samples to be scored. Besides, when the current time slot exceeds the maximum frame number, which can be scored, the sample with the lowest score will be discarded. Consequently, the training-set can be efficiently and reliably distilled. Comprehensive tests on two well-known UAV benchmarks prove the effectiveness of our method with real-time speed on a single CPU.

I. INTRODUCTION

With the development of unmanned aerial vehicle (UAV), visual tracking plays an increasingly important role in prosperous practical applications, such as obstacle avoidance [1], aerial refueling [2], autonomous landing [3], etc. Appreciable progress has been made in UAV tracking in recent years. However, it still faces many challenges, including strong UAV/object motion, frequent viewpoint change, severe illumination variation, abnormal appearance variation (occlusion or out-of-view). Additionally, mobile aerial platform has increased the difficulty in tracking scenarios such as mechanical vibration, restricted computational capability, limited power capacity, to name a few.

In recent years, correlation filter (CF)-based approaches [4] have become a widely used framework for visual tracking due to its outstanding computational efficiency obtained in Fourier domain, especially for aerial tracking tasks in which the computational resources are precious. The training sample set used to learn the CF model is collected from the video frames. In traditional CF-based methods, to help CF adapt to the object appearance change, the online learned CF model is updated with new samples frame-by-frame via a simple linear interpolation method. Unfortunately, this strategy can be easily affected by unreliable samples, which can be introduced by many challenges, e.g., occlusion and out-of-view, leading to suboptimal performance, as shown in Fig. 1.

Existing CF-based trackers either ignore the problem mentioned above [6]–[10] or directly adjust the learning rate for model update if the new samples are criticized as unreliable [11]–[13]. An efficacious approach is to manage training-set via explicit component [14], [15]. However, they have to process large prior samples in each frame, leading to an undesired increase in the computational burden.

In the online tracking process, when a new frame arrives, CF-based tracker firstly obtains a response map by correlating the learned filter and new samples, and then the object location is predicted according to the peak in the map [16]–[18]. The quality of the response map, such as the sharpness and the fluctuation, can to some extent reflect the confidence degree about the tracking result [7]. In this work, the response map is exploited to measure the reliability of training samples. Frames with high confident tracking results are used as keyframes to divide the tracking into multiple time slots. Besides, the response map is also integrated into a joint correlation filter and sample weight optimization framework, which is developed to score the training sample adaptively. To speed up the scoring process, the training-set

1Fan Li, Changhong Fu, Fuling Lin and Yiming Li are with the School of Mechanical Engineering, Tongji University, 201804 Shanghai, China. changhongfu@tongji.edu.cn
2Peng Lu is with the Adaptive Robotic Controls Lab (ArcLab), Hong Kong Polytechnic University (PolyU), Hong Kong, China. peng.lu@polyu.edu.hk
The source code and UAV tracking videos are available in https://github.com/vision4robotics/TSD-Tracker and https://youtu.be/2ZlDYrq2FBA.
will be fused into one key-sample once the new time slot is established. Besides, if the capacity of training-set is larger than the scoring capacity, the samples with the lowest score will be discarded. Our core contributions are:

- A novel time slot-based distillation algorithm for UAV tracking (TSD) is proposed to enhance the quality of the training-set efficiently.
- Extensive and comprehensive tests on 193 challenging UAV image sequences have demonstrated that TSD tracker achieves competitive performance against the state-of-the-art works and runs at real-time frame rates.

II. RELATED WORKS

Discriminative correlation filter-based tracker was firstly introduced in the minimum output sum of squared error (MOSSE) [12] filer. J. F. Henriques et al. [4] further introduced the kernel trick into CF-based framework. Computing efficiency was improved greatly by employing the circular matrix property and solving the regression problem in the Fourier domain. By introducing modern multi-dimensional features, the accuracy of CF-based approaches is further improved [19]–[21]. However, as the CF-based trackers usually obtain negative samples by shifting images cyclically, the trained filter is influenced undesirably by boundary effect. Existing approaches [22]–[24] have made up for this shortcoming by spatial regularization. To utilize real background information, learning background-aware correlation filters (BACF) [5] enlarged search regions to extract real negative training examples from the background. However, those methods are susceptible to unreliable samples which were introduced by occlusion, out of view, viewpoint change and other reasons.

Most existing methods choose not to update the model if the samples do not meet certain criteria. D. S. Bolme et al. [12] and M. Wang et al. [11] rejected new samples based on the peak-to-sidelobe ratio (PSR) and peak-to-correlation energy (APCE), respectively. However, due to some challenging scenarios like viewpoint change, some samples that were previously considered reliable may turn into unreliable samples, as shown in the middle row of Fig. 6. Those samples may lead to model drift. Other methods exploit explicit component to manage training-set. In [14], an explicit component based on distance comparisons is used to manage the training-set, and J. Zhang et al. [15] used a combination of experts to correct undesirable model updates. A unified formulation for discriminative tracking [25] evaluated the quality of the samples to manage the training-set dynamically. This type of method can improve the quality of the training-set efficaciously. However, they have to process large samples, thereby struggling to meet real-time performance requirements for UAV tracking scenarios.

III. PROPOSED TRACKING APPROACH

In this section, the baseline BACF is firstly reviewed for better understanding, then our TSD tracker is presented. The main symbols in this work are summarized in Table I.

A. Revisiting BACF

Given the $d$-th channel of the vectorized training samples $x_{[d]}$ and predetermined vectorized ideal response $y_{[d]} \in \mathbb{R}^N$, the objective of training the filter $w$ is to minimize:

$$
E(w) = \sum_{j=1}^{N} \left| \sum_{d=1}^{D} w_{d}[\Delta \tau_j] x_{[d]} \right|^2 + \frac{\lambda}{2} \sum_{d=1}^{D} \left| w_{d} \right|^2,
$$

where $[\Delta \tau_j]$ is the circular shift operator, and $x_{[d]}[\Delta \tau_j]$ denotes $j$-step circular shifted sample $x_{[d]}$. $w_{[d]} \in \mathbb{R}^M$ denotes the learned correlation filter. B is a $M \times N$ binary matrix which crops the mid $D$ elements of sample $x_{[d]}$. It is worth noting that $N > M$.

Though BACF has achieved satisfactory performance owing to the augment of real-world negative samples, it still adopts a simple linear interpolation method to update the training samples, raising the possibility of introducing unreliable information like occlusion or outdated appearance. In this work, a time slot-based joint filter and sample score optimization framework is proposed built on BACF. Specifically, the training-set is restricted to a small size for raising the optimization speed. When the number of current training samples exceeds the given size, the most unreliable sample will be discarded. In addition, frames with high confident tracking results are employed as the keyframes to divide the tracking process into multiple time slots. Once the slot is established, the samples are fused into one key-sample with their weights.

B. Overall objective of TSD

In this work, a novel dynamic time slot-based distillation approach is constructed. The proposed filter $w$ and sample scores $\alpha$ can be collaboratively learned by minimizing a cooperative energy minimization function as follows:

$$
E(w, \alpha) = E_1(w, \alpha) + E_2(\alpha) + E_3(\alpha),
$$

where each term is described below.

1) Classification error $E_1$:

Similar to BACF, given a training-set $\{x^1, x^2, ..., x^F\}$, the loss for the discrepancy between the scheduled response and the filter response for
the sample $x^f$ is defined as:

$$E_1(w, \alpha) = \sum_{f=1}^{F} \left( \alpha^f \sum_{j=1}^{N} \|y_j - \sum_{d=1}^{D} \mathbf{w}_{[d]}^T \mathbf{B} x_{[d]}^f \Delta \tau_j^f \|_2^2 \right) + \frac{\lambda}{2} \sum_{d=1}^{D} \|w_d\|_2^2,$$  

where $x^f$ is the $f$-th sample in the training-set, $\alpha^f$ is the score of sample $x^f$, and $\lambda$ is a regularization parameter.

Different from BACF which sets the score $\alpha^f \in \mathbb{R}$ using a fixed learning rate parameter, in TSD, different samples have distinct scores based on their confidence. Reliable samples have higher scores, and vice versa. The number of training samples $F$ is defined by:

$$F = \min \{ f, F_{\text{max}} \},$$

where $f$ denotes the frame number in the current time slot. To avoid exceeding the upper limit of memory consumption, the training-set capacity $F_{\text{max}}$ is proposed. If the number of samples exceeds $F_{\text{max}}$, the sample with the lowest score will be discarded.

2) Temporal regularization $E_2$: To account for fast appearance changes, recent samples are given larger scores:

$$E_2(\alpha) = \frac{\gamma}{2} \sum_{f=1}^{F} \left( \frac{\alpha^f}{t^f} \right)^2,$$

where $\gamma$ is a trade-off parameter between classification error and temporal regulation. $t^f$ is a function related to the distance between the current frame and the last selected keyframe for time slot division. It is designed as follows:

$$t^f = \begin{cases} a^{-1}, & f = 1, \ldots, F - f_0 \\ a^{-1}(1 - q)^{F - f_0 - f}, & f = F - f_0 + 1, \ldots, F \end{cases},$$

where the constant $a = F - f_0 + \frac{(1 - q)^{F - f_0 - 1}}{q}$ is determined by the condition $\sum_{f=1}^{F} t^f = 1$. In this work, $f_0$ and $q$ are set to 10 and 0.0408, respectively.

3) Response map-based regularization $E_3$: The confidence degree about the tracking result can be reflected by the quality of the response map to some extent. Thus the samples with higher quality responses are given higher scores:

$$E_3(\alpha) = \frac{\nu}{2} \sum_{f=1}^{F} \frac{(\alpha^f)^2}{\lambda (FMR)^2},$$

where $\nu$ is a trade-off parameter between classification error and response map-based regulation. In this work, the quality of the response map is evaluated by the proposed dual-area peak to media ratio ($\lambda (FMR)$) that is described in Section III-D.

C. Optimization algorithm

To unify the dimensions of each term in Eq. (2), intermediate variable $h_{[d]}$ is introduced as follows:

$$h_{[d]} = [0, w_{[d]}^T, 0]^T.$$  

Thus Eq. (2) can be expressed in the frequency domain as:

$$E = \sum_{f=1}^{F} \left( \alpha^f \|g_{u,[d]} - \sum_{d=1}^{D} \hat{g}_{u,[d]}^* \hat{x}^f_{u,[d]} \|_2^2 \right) + \frac{\lambda}{2} \| \hat{h}_{u,[d]} \|_2^2,$$

$$+ \frac{2}{2} \sum_{f=1}^{F} \frac{(\alpha^f)^2}{t^f} + \frac{\nu}{2} \sum_{f=1}^{F} DPMR' (\alpha^f)^2,$$

s.t. $\hat{g}_{u,[d]}^* = \hat{h}_{u,[d]}^*$

where * denotes the Discrete Fourier Transform (DFT) of a signal, and * denotes complex conjugate. As for the subscript $u$, it represents the element in a data set (i.e., $\hat{x}_u$ refers to the element in a training-set). Different from BACF, all the operations in Eq. (9) are performed in element-wise.

To learn parameter $\hat{w}^*$ and $\alpha$, i.e., for learning parameter $\hat{h}^*$ and $\alpha$, the problem is how to optimize Eq. (9). Two components of parameter can be optimized iteratively, i.e., parameter $\hat{h}^*$ trained with fixed $\alpha$ firstly and then the process is inverted.

1) Subproblem $\hat{h}^*$: $\hat{h}^*$ can be solved using Augmented Lagrangian Method (ALM):

$$E = \sum_{f=1}^{F} \left( \alpha^f \|g_{u,[d]} - \sum_{d=1}^{D} \hat{g}_{u,[d]}^* \hat{x}^f_{u,[d]} \|_2^2 \right) + \frac{\lambda}{2} \| \hat{h}_{u,[d]} \|_2^2,$$

$$+ \frac{2}{2} \sum_{f=1}^{F} \frac{(\alpha^f)^2}{t^f} + \frac{\nu}{2} \sum_{f=1}^{F} DPMR' (\alpha^f)^2,$$

where $\mu$ and $\zeta$ denote the penalty factor and the element of the Lagrangian vector in the Fourier domain separately. With the ADMM [26] technique, Eq. (10) can be solved as follows:

$$\hat{h}_{u,[d]} = \arg \min_{\hat{h}_{u,[d]}} \left\{ \frac{\lambda}{2} \| \hat{h}_{u,[d]} \|_2^2 + \zeta^T (\hat{g}_{u,[d]} - \hat{h}_{u,[d]}^*) + \frac{\mu}{2} \| \hat{g}_{u,[d]} - \hat{h}_{u,[d]}^* \|_2^2 \right\},$$

$$\hat{g}_{u,[d]} = \arg \min_{\hat{g}_{u,[d]}} \left\{ \sum_{f=1}^{F} \alpha^f \| g_{u,[d]} - \hat{g}_{u,[d]} \|_2^2 \right\} + \zeta^T (\hat{g}_{u,[d]} - \hat{h}_{u,[d]}^*) + \frac{\mu}{2} \| \hat{g}_{u,[d]} - \hat{h}_{u,[d]}^* \|_2^2 \right\},$$

Thanks to the element-wise operation, the solution to both subproblems $\hat{g}^*$ and $\hat{h}^*$ can be easily obtained as follows:

$$\hat{h}_{u,[d]} = (\lambda + \mu)^{-1} \left( 2\zeta^T + \mu \hat{g}_{u,[d]} \right)^{-1},$$

$$\hat{g}_{u,[d]} = \left( \sum_{f=1}^{F} \alpha^f \| \hat{x}^f_{u,[d]} \|_2^2 \right)^{-1} \left( \sum_{f=1}^{F} \alpha^f \hat{g}_{u,[d]} \hat{x}^f_{u,[d]} - \zeta^T + \frac{\mu}{2} (\hat{g}_{u,[d]})^* \right),$$

The Lagrangian parameter in Eq. (10) is updated as follows:

$$\zeta^{(j+1)}_{u,[d]} = \zeta^{(j)}_{u,[d]} + \mu \left( \hat{g}_{u,[d]}^{(j+1)*} - \hat{h}_{u,[d]}^{(j+1)*} \right),$$

where superscript $(j)$ denotes the initial value or the value in the last iteration, and subscript $(j + 1)$ denotes the value.
Fig. 2. Visualization of the calculation of $DPMR$. The first column are images from example sequence person17_1. The response maps in the second column come from TSD. The third column are the high and the low response area split from the response maps in (d).

at current iteration. Note that the subscript $u$ refers to the iterations of variable, and $h_{u[d]} = [0, w_{u[d]}, 0]^T$.

2) Subproblem $\alpha$: The second step of optimization is to train $\alpha$ with fixed $h^*$, which requires setting the first derivative of $\alpha$ to zero. Therefore, $\alpha$ can be obtained by:

$$
\frac{\partial J}{\partial \alpha} = - \frac{\partial}{\partial \alpha} \left( \sum_{f=1}^{F} \left( \alpha^f \right) \left\| \tilde{y}_{u[d]} - \sum_{d=1}^{D} (g_{u[d]}^f) x_{u[d]}^f \right\|^2_2 \right) + \frac{\gamma}{2} \sum_{f=1}^{F} \left( \alpha^f \right)^2 + \nu \sum_{f=1}^{F} DPMR^f (\alpha^f)^2,
$$

(14)

the above subproblem is equivalent to the quadratic programming problem as follows:

$$
\min \ J = \sum_{f=1}^{F} (\beta^f \alpha^f + \gamma^f (\alpha^f)^2) \\
\text{s.t.} \sum_{f=1}^{F} \alpha^f = 1
$$

(15)

This optimization problem is a convex quadratic programming method. Therefore, $\alpha$ can be solved efficiently via standard quadratic programming.

D. Time slot establishment

The quality of the response map can indicate the reliability of the tracking result. As illustrated in (b) and (d) of Fig. 2, the response map with only one sharp peak is of high quality, and it should be smooth in the other areas. As for the tracking result with a low confidence degree, the response map will fluctuate intensely as shown in (a) and (c) of Fig. 2. Traditional CF-based trackers use the peak value of the response map to evaluate its quality, which is inaccurate in many cases as shown in Fig. 2. In this work, the quality of the response map is evaluated by the dual-area peak to media ratio ($DPMR$).

To calculated the $DPMR$, the response map is split into the high response area and the low response area as shown in (c) and (f) of Fig. 2. The $DPMR$ is then defined as:

$$
DPMR = \frac{\max(R_h) - \min(R_l)}{\text{mean}(R_l) - \min(R_l)},
$$

(16)

where $R_h$ and $R_l$ denote the high and low response area, respectively. $DPMR$ can indicate the fluctuated level of response map, which can reflect the confidence degree of the detected target. Finally, our strategy is designed as:

$$
\text{breakpoint} (M) = \begin{cases} 
1, & DPMR > tr \\
0, & \text{otherwise}
\end{cases}
$$

(17)

In practice, we choose $tr = 14$ based on our empirical results. If the output of breakpoint is 1, this frame will be treated as keyframe to divide the tracking process, and a new time slot is established. Then this slot is fused to one key-sample.

E. Weighted fusion

As mentioned above, the sample score $\alpha$ can evaluate the quality of each sample. To decrease the number of samples to be scored, at the end of a period of time slot, the last distilled training-set is fused into one key-sample $x_{key}$ as follows:

$$
x_{key} = \sum_{f=1}^{F} (\alpha^f x^f),
$$

(18)

where $x^f$ is the $f$-th sample in the last distilled training-set. $\alpha^f$ is the score of the sample in $f$-th selected frame. Key-sample is taken as the first sample of the next time slot.

IV. EXPERIMENTS

In this section, the proposed TSD tracker is evaluated on 193 challenging UAV image sequences from two well-known and frequently-used benchmarks, i.e., UAV123@10fps [32] and DTB70 [33]. The experimental results are compared to state-of-the-art trackers with real-time speed on a single CPU (>30fps), i.e., DSST [19], BACF [5], Staple [28], Staple_CA [9], MCCT-H [29], ECO-HC [24], fDSST [30], DCF [4], KCF [4], and KCC [31].

A. Implementation details

TSD tracker is implemented in Matlab R2018a. Color names (CN) [20] is adopted as the feature representation to raise efficiency. The training-set capacity $F_{max}$ in Eq. (4) is set to 50. $\gamma$ in Eq. (5) and $\nu$ in Eq. (7) are set to 3.02 and 0.201. In Eq. (6), $f_0 = 10$ and $q = 0.0408$. For ADMM iteration, $g$, $w$ and $h$ are all initialized using null matrices, and update scale is set to 2. All the experiments are run on the computer with an i7-8700K (3.70KHz) CPU, 32GB RAM, and a single NVIDIA RTX 2080 GPU for fair comparisons.

B. Comparison with real-time trackers on a single CPU

Due to the extremely harsh computational resources onboard UAVs, an energy-efficient and low-cost CPU is desirable in UAV tracking applications. In this work, we evaluate the trackers with real-time frame rates on a single CPU.
TSD outperforms the second-best tracker ECO-HC and third-best tracker MCCT-H by 2.2% and 8.8% in the precision, respectively. In success plots, TSD is ranking No.2 among 11 state-of-the-art trackers. Yet it is noted that the best tracker ECO-HC [24] employs both histogram of gradient (HOG) [4] and CN. Besides satisfactory tracking results, the speed of TSD is adequate for real-time UAV tracking applications, as shown in Table II.

2) Attribute based comparison: Besides overall performance, the attribute-based performance of TSD and other trackers are also evaluated. Precision plots in the scenarios of partial occlusion, viewpoint change, out of view, aspect ratio change, camera motion, fast motion, full occlusion, illumination variation as well as similar object around are demonstrated in Fig. 5. TSD has exhibited a huge improvement from its baseline BACF, and has achieved state-of-the-art performance in all the challenging attributes. In these scenarios, unreliable samples can be easily introduced into training-set. Typically, CF-based trackers ignore to manage unreliable samples. Therefore, TSD is able to improve the quality of the training-set and reduce the unexpected effects caused by unreliable samples efficiently.

![Precision plots on UAV123@10fps](image1)

![Success plots on UAV123@10fps](image2)

![Precision plots on DTB70](image3)

![Success plots on DTB70](image4)

**Fig. 3.** Precision and success plots of TSD as well as other state-of-the-art real-time trackers on a single CPU. Two standard evaluation measures are employed in precision and success plots, i.e., center location error (CLE) and success rate based on one-pass evaluation (OPE) [27]. The CLE is defined as the Euclidean distance between the center of the estimated bounding box and ground-truth location, which can measure the precision. The success rate is characterized as the intersection over union (IoU) of the tracker bounding box and ground-truth bounding box, which can indicate the precision of scale estimation.

**TABLE II**

| Algorithms | TSD  | DSST  | Staple  | Staple CA  | MCCT-H  | ECO-HC  | IDSST  | KCC  | DCF  | KCF  | FPS  | MSPF  |
|------------|------|-------|---------|------------|---------|---------|--------|------|------|------|------|-------|
|            |      |       |         |            |         |         |        |      |      |      |      |       |
|            | 0.659 | 0.642 | 0.420   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.466 | 0.468 | 0.572 | 0.587 |
|            | 0.657 | 0.643 | 0.433   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.468 | 0.468 | 0.572 | 0.587 |

![Overall performance on UAV123@10fps](image5)

**Fig. 4.** Ranking map of the real-time state-of-the-art trackers on CPU.

1) Overall performance evaluation: The experimental results achieved by TSD tracker and other state-of-the-art trackers on UAV123@10fps [32] and DTB70 [33] are demonstrated in Fig. 3, in which precision plots (PPs) and success plots (SPs) are employed for evaluation. As shown in the precision plots, TSD outperforms the second-best tracker ECO-HC and the third-best tracker MCCT-H by 3.9% and 10.6%, respectively. Similar to PPs, TSD is also ranking No.1 among other state-of-the-art trackers in SPs. Ranking map on UAV123@10fps in terms of PPs and SPs is displayed in Fig. 4. As for DTB70, TSD outperforms the second-best tracker ECO-HC and third-best tracker MCCT-H by a gain of 2.2% and 8.8% in the precision, respectively. In success plots, TSD is ranking No.2 among 11 state-of-the-art trackers. Yet it is noted that the best tracker ECO-HC [24] employs both histogram of gradient (HOG) [4] and CN. Besides satisfactory tracking results, the speed of TSD is adequate for real-time UAV tracking applications, as shown in Table II.

**TABLE II**

| Algorithms | TSD  | DSST  | Staple  | Staple CA  | MCCT-H  | ECO-HC  | IDSST  | KCC  | DCF  | KCF  | FPS  | MSPF  |
|------------|------|-------|---------|------------|---------|---------|--------|------|------|------|------|-------|
|            |      |       |         |            |         |         |        |      |      |      |      |       |
|            | 0.659 | 0.642 | 0.420   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.466 | 0.468 | 0.572 | 0.587 |
|            | 0.657 | 0.643 | 0.433   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.468 | 0.468 | 0.572 | 0.587 |

**Fig. 3.** Precision and success plots of TSD as well as other state-of-the-art real-time trackers on a single CPU. Two standard evaluation measures are employed in precision and success plots, i.e., center location error (CLE) and success rate based on one-pass evaluation (OPE) [27]. The CLE is defined as the Euclidean distance between the center of the estimated bounding box and ground-truth location, which can measure the precision. The success rate is characterized as the intersection over union (IoU) of the tracker bounding box and ground-truth bounding box, which can indicate the precision of scale estimation.

**TABLE II**

| Algorithms | TSD  | DSST  | Staple  | Staple CA  | MCCT-H  | ECO-HC  | IDSST  | KCC  | DCF  | KCF  | FPS  | MSPF  |
|------------|------|-------|---------|------------|---------|---------|--------|------|------|------|------|-------|
|            |      |       |         |            |         |         |        |      |      |      |      |       |
|            | 0.659 | 0.642 | 0.420   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.466 | 0.468 | 0.572 | 0.587 |
|            | 0.657 | 0.643 | 0.433   | 0.379      | 0.333   | 0.465   | 0.456  | 0.466 | 0.468 | 0.468 | 0.572 | 0.587 |

![Overall performance on UAV123@10fps](image6)

**Fig. 4.** Ranking map of the real-time state-of-the-art trackers on CPU.
3. Ablation study

The ablation study is conducted on UAV123@10fps [32]. Significant performance improvement is achieved by integrating the unreliable sample discarding with the baseline. As shown in Table III, it outperforms the baseline in precision plots and success plots by a gain of 11.0% and 7.7%, respectively. The weighted sample fusion based on the time slot establishment further improves the performance by reducing the influence of untrustworthy samples. Additionally incorporating the proposed response map based regularization has elevated TSD to 0.659 and 0.464 in precision and success rate respectively, leading to a final improvement of 15.2% and 12.3% respectively compared to the baseline.

V. Conclusions

In this work, a novel correlation filter with training-set distillation is proposed for UAV tracking. In the process of training-set distillation, historical samples are scored dynamically for enhancing the tracking reliability. For efficiency reason, the proposed tracker has employed keyframes to divide the tracking process into multiple time slots. In the current time slot, the most unreliable sample will be discarded when the number of current training samples exceeds the given size. Samples in the newly established time slot are fused into one sample immediately to decrease the training redundancy. Extensive tests have validated that our tracker outperforms significantly better than many state-of-the-art works. We believe that our method can further improve the development of UAV tracking.

ACKNOWLEDGMENT

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

[1] C. Fu, A. Carrio, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, “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.

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

[3] S. Lin, M. A. Garratt, and A. J. Lambert, “Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment,” Autonomous Robots, vol. 41, pp. 881–901, 2017.

[4] J. F. Henriquez, R. Caceres, P. Martins, and J. Batista, “High-speed tracking with normalized correlation filters,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, pp. 583–596, 2015.

[5] 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.

[6] A. Lukezic, T. Vojir, L. Čehovin Zajc, J. Matas, and M. Kristan, “Discriminative correlation filter with channel and spatial reliability,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6309–6318.

[7] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, “Learning aberrance repressed correlation filters for real-time uav tracking,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2891–2900.

[8] L. Zhang and P. N. Suganthan, “Robust visual tracking via co-trained Kernelized correlation filters,” Pattern Recognition, vol. 69, pp. 82–93, 2017.

[9] 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. 1396–1404.

[10] C. Fu, F. Lin, Y. Li, and G. Chen, “Correlation Filter-Based Visual Tracking for UAV with Online Multi-Feature Learning,” Remote Sensing, vol. 11, pp. 1–23, 2019.

[11] M. Wang, Y. Liu, and Z. Huang, “Large margin object tracking with circulant feature maps,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4021–4029.

[12] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2544–2550.

[13] T. Zhang, C. Xu, and M.-H. Yang, “Multi-task correlation particle filter for robust object tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4335–4343.

[14] J. Gao, H. Ling, W. Hu, and J. Xing, “Transfer learning based visual tracking with gaussian processes regression,” in Proceedings of European Conference on Computer vision (ECCV), 2014, pp. 188–203.

[15] J. Zhang, S. Ma, and S. Sclaroff, “MEEM: robust tracking via multiple experts using entropy minimization,” in Proceedings of European Conference on Computer Vision (ECCV), 2014, pp. 188–203.

[16] Y. Li and J. Zhu, “A scale adaptive kernel correlation filter tracker with feature integration,” in Proceedings of European Conference on Computer Vision (ECCV) Workshops, 2014, pp. 254–265.

[17] J. Johander, M. Danelljan, F. S. Khan, and M. Felsberg, “DCCO: Towards deformable continuous convolution operators for visual tracking,” in Proceedings of International Conference on Computer Analysis of Images and Patterns (CAIP), 2017, pp. 55–67.

[18] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in Proceedings of European Conference on Computer Vision (ECCV), 2016, pp. 472–488.

[19] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” in Proceedings of British Machine Vision Conference (BMVC), 2014.

[20] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, “Adaptive color attributes for real-time visual tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1090–1097.

[21] H. Fan and H. Ling, “Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking,” in Proceedings of IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5486–5494.

[22] M. Danelljan, G. Häger, F. Shahbaz 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.

[23] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, “Learning spatial-temporal regularized correlation filters for visual tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4904–4913.

[24] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, “ECO: efficient convolution operators for tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6638–6646.

[25] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, “Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1430–1438.

[26] 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.

[27] 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.

[28] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, “Staple: Complementary learners for real-time tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1401–1409.

[29] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, “Multi-cue correlation filters for robust visual tracking,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4844–4853.

[30] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, “Discriminative scale space tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1561–1575, 2017.

[31] C. Wang, L. Zhang, L. Xie, and J. Yuan, “Kernel cross-correlator,” in Proceedings of AAAI Conference on Artificial Intelligence, 2018, pp. 4179–4186.

[32] 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.

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