FRATCF: FEATURE-RESIDUE REAL-TIME UAV TRACKING BASED ON AUTOMATIC SPATIO-TEMPORAL REGULARIZATION CORRELATION FILTER

Yuan Yuan, Yanming Chen, Yueqing Jing, Peng Zhou, Yiwen Zhang*

School of Compute Science and Technology, Anhui University, Hefei, China
{cym, zhoupeng, zhangyiwen}@ahu.edu.cn, {e20201126, jingyueqing}@stu.ahu.edu.cn

2022 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-8563-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICME52920.2022.9859881

ABSTRACT

Traditional discriminative correlation filter (DCF) has received widespread popularity due to its high computational efficiency. However, most of the existing DCF-based trackers improve the learning of the target object by introducing some simple regularization methods in the detection stage, which may easily lose the tracking target in scenes with background clutter, fast-moving cameras and similar targets. We propose a feature residual filter with automatic spatio-temporal regularization, namely FRATCF, which can be strengthened the filter learning by introducing the feature residual between two adjacent frames in the training phase. Extensive experiments are conducted on two challenging unmanned aerial vehicle (UAV) benchmarks, i.e., UAV123@10fps and DTB70. Results prove that our tracker runs at $\sim$43 FPS on an extremely cheap configuration, which is about twice the speed of AutoTrack. The performance is also better than other state-of-the-art (SOTA) trackers.

***Index Terms*—** UAV tracking, feature residual, correlation filter, automatic spatio-temporal regularization

1. INTRODUCTION

As a hot spot in the field of computer vision, visual tracking has received widespread attention. Visual object tracking is one of the fundamental tasks in the computer vision community, aiming to estimate the position and scale of the target in the subsequent frames of the image sequence only with the information given in the first frame. In recent years, when many researchers applied tracking algorithms to UAV platforms, tracking algorithms opened up another wave of novel applications, such as traffic monitoring [1], aerial cinematography [2] and disaster surveillance [3]. However, UAVs still have certain challenges in visual tracking, mainly due to three reasons: (a) The background of UAV tracking is very complicated, and the target is also prone to visual occlusion. In addition, the UAV can fly around the object to collect the moving image of the target. These will cause a certain deviation and make the UAV tracking more difficult. (b) When UAV tracking, the onboard power capacity and computing resources are very limited. Most UAVs only use a single CPU as the computing platform, which greatly limits the processing speed. (c) The target tracked by UAV has real-time requirements. During the entire tracking process, we need to observe the video stream returned by the UAV camera in real time through the ground station. Therefore, the tracking algorithm must be lightweight in order to process the data faster for the purpose of real-time UAV tracking. Because of the difficulties caused by the characteristics of UAVs, we need to get more efficient algorithms to design a robust UAV target tracker.

In order to solve some of the difficulties in UAV tracking, two major research branches have emerged in recent years. One is a tracking algorithm based on discriminative correlation filters [4–7], and the other is based on deep learning or tracking algorithm with depth features [8, 9]. Among them, the use of deep convolution features (CNN) requires the use

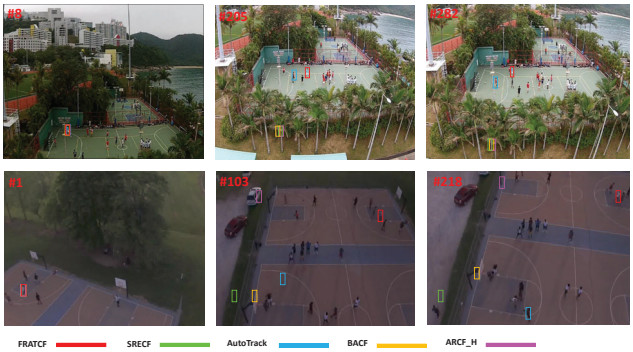


Fig. 1: Presentation of the tracking results in FRATCF (red rectangle), the other three SOTA trackers (SRECF (green rectangle), AutoTrack (blue rectangle), ARCF_H (purple rectangle)) and the baseline BACF tracker (orange rectangle) on the sequences of Basketball and StreetBasketball1. As shown above, when there are background clutter, similar targets and fast camera movement in the UAV tracking scene, the other 4 SOTA trackers almost completely lose the target, but the FRATCF (red rectangle) can accurately track the target.

*Corresponding Author: Yanming Chen(cym@ahu.edu.cn)

This work is supported in part by the National Natural Science Foundation of China under Grand (61802001) and the Key Natural Science Foundation of Education Department of Anhui province (KJ2021A0046).

of GPU. Although it improves some accuracy, it not only hinders the tracking speed, but also requires a high price. It is not practical in real-time tracking scenes. Blome et al. [10] proposed MOSSE, which used correlation filters for visual tracking for the first time. Henriques et al. [11] proposed KCF filter, by applying to multiple channel features, and redefining the objective function of MOSSE [10] as ridge regression, using the DCF framework for Fourier domain calculations, greatly improving efficiency. Due to filter boundary effects, appearance changes, target drift and other issues, most well-known trackers introduce regularization terms to improve performance. Danelljan et al. [5] proposed SRDCF, which solves the boundary effect problem by performing spatial penalty on DCF coefficients, which inevitably improves tracking performance at the cost of increased complexity. Li et al. [12] proposed STRCF by combining time and space regularization at the same time. Huang et al. [6] proposed ARCF, which used the response graph generated in the detection stage to restrict target learning. In recent years, Li et al. [13] further proposed the AutoTack filter, which maked full use of local-global response changes to train its tracker, and achieved an online automatic spatio-temporal regularization performance. However, they may lost the tracking target in scenes with background clutter, fast-moving cameras and similar targets. As shown in Fig.1, it can be seen that when the above scenes occur, the other SOTA trackers almost completely lose the target, but the FRATCF (red rectangle) can accurately track the target. In visual tracking, residual learning was applied to capture the difference between the base layer output and the ground truth to reduce model degradation during online update. Residuals can be the difference between the target and the model output, the difference between the input and the output, the difference between a sample and the sample mean. Since the video is approximately time continuous, residuals are usually small values and entropy, which can be used to speed up the learning of correlation filters. The existing filters for residual learning [14, 15] do not allow the parameters to be updated adaptively. In this paper, a residual is formed by subtracting the feature of the current frame from that of the previous one, which can be used to speed up the learning of the filter. Fig.2 is the tracking framework of the FRATCF we proposed.

Our main contributions are summarized as follows:

- A novel feature residual method, supported by the automatic spatio-temporal regularization, is proposed. The proposed FRATCF can strengthen the filter learning by introducing the feature residuals between two adjacent frames in the training phase.
- In order to effectively solve FRATCF, we use ADMM algorithm to simplify the calculation. After a series of derivations, each sub-problem of the algorithm has a closed solution. Our algorithm can empirically converge in very few iterations.
- The proposed FRATCF tracker is exhaustively tested on two challenging UAV benchmarks. The performance is better than other SOTA trackers. Our tracker runs at $\sim$43 FPS in an extremely cheap configuration, which is about twice the speed of AutoTrack tracker, and is sufficient for real-time UAV applications.

2. METHOD

2.1. Objective function of FRATCF

We select the BACF [4] tracker as our baseline. Given the extracted feature $x_t^d \in \mathbb{R}^{N\times 1}(d = 1, 2, \cdots, D)$ with length N in frame t, where D denotes number of channel, and the desired Gaussian-shaped response $y \in \mathbb{R}^{N\times 1}$, the desired filter $h_t = \left[h_t^1, h_t^2, \cdots, h_t^D\right] \in \mathbb{R}^{N\times D}$ can be optimized through minimizing the following objective function:

$$\begin{aligned} G\left(h_t, E_t\right) = & \frac{1}{2}\left\| y - \sum_{d=1}^{D} x_t^d * C h_t^d \right\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{D}\left\| h_t^d \right\|_2^2 \\ & + \frac{\gamma}{2}\left\| \sum_{d=1}^{D} \xi_t * C h_t^d \right\|_2^2 + \frac{1}{2}\sum_{d=1}^{D}\left\| \alpha \odot h_t^d \right\|_2^2 \\ & + \frac{E_t}{2}\sum_{d=1}^{D}\left\| h_t^d - h_{t-1}^d \right\|_2^2 + \frac{1}{2}\left\| E_t - e \right\|_2^2 \end{aligned} \tag{1}$$

where $\xi_t^d = x_t^d - x_{t-1}^d$, E_t denotes the optimized temporal regularization parameter, $C \in \mathbb{R}^{N\times M}(M \ll N)$ represents a binary matrix to crop the mid M elements of x_t^d, λ is a penalty coefficient, γ is the sensitivity factor, which controls the filter's sensitivity to feature-residue, $*$ denotes the correlation operator, $\odot$ denotes the Hadamard product. α and e respectively denote the automatic spatial regularization parameter and reference regularization parameter, which are calculated by AutoTack [13].

2.2. Optimization with ADMM

Using an auxiliary variable $\hat{v}_t = \left(I_D \otimes FC\right) h_t$, where $I_D \in \mathbb{R}^{D\times D}$ is an identity matrix, $F \in \mathbb{R}^{N\times M}$, the Fourier form of Eq.(1) can be written as $G\left(h_t, E_t, \hat{v}_t\right)$[13, 15]. Then, we employ an Augmented Lagrangian Method(ALM)[4]:

$$\begin{aligned} \mathcal{L}_t\left(h_t, E_t, \hat{v}_t, \hat{\varepsilon}\right) = & G\left(h_t, E_t, \hat{v}_t\right) + \frac{\mu}{2}\left\| \hat{v}_t - \left(I_D \otimes FC\right) h_t \right\|_2^2 \\ & + \hat{\varepsilon}^{\mathrm{H}}\left(\hat{v}_t - \left(I_D \otimes FC\right) h_t\right) \end{aligned} \tag{2}$$

where μ is the the step size regularization parameter, $\hat{\varepsilon} = \left[\hat{\varepsilon}_1^{\mathrm{H}}, \ldots, \hat{\varepsilon}_D^{\mathrm{H}}\right]^{\mathrm{H}}$ is the Lagrangian vector of size $DN \times 1$ in the frequency domain, $\hat{v}_t = \left[\left(\hat{v}_t^1\right)^{\mathrm{H}}, \ldots, \left(\hat{v}_t^D\right)^{\mathrm{H}}\right]^{\mathrm{H}}$, the symbol $\hat{}$ denotes the discrete Fourier transform (DFT) of a signal, the operator $^{\mathrm{H}}$ computes the conjugate transpose on a complex vector or matrix, $\otimes$ indicates the Kronecker product.

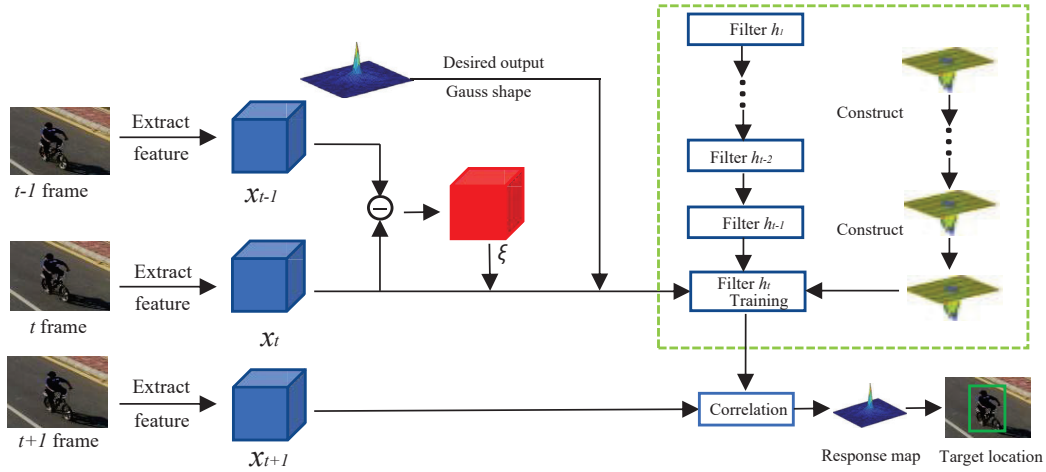


Fig. 2: Main workflow of the proposed FRATCF. In the training phase, we subtract the features that are extracted from the t and $t-1$ frames to obtain a residual value, and update the global-local automatic spatio-temporal regularization parameters to strengthen the training of the filter. When we get the updated value of the filter and the features extracted from the next frame, we can locate the target position.

Eq.(2) can be solved iteratively using the ADMM technique[4]. Fortunately, close form solutions can be found for following three subproblems, h_t^*, $\hat{v}_t^*$ and E_t^*.

Solution to subproblems h_t^* and $\hat{v}_t^*$:

$$\begin{aligned} h_t^* = \underset{h_t}{\operatorname{argmin}} \Big\{ & \frac{1}{2}\left\|\alpha \odot h_t\right\|_2^2 + \frac{E_t}{2}\left\|h_t - h_{t-1}\right\|_2^2 \\ & + \frac{\lambda}{2}\left\|h_t\right\|_2^2 + \frac{\mu}{2}\left\|\hat{v}_t - (I_D \otimes FC)\, h_t\right\|_2^2 \\ & + \hat{\varepsilon}^{\mathrm{H}}\left(\hat{v}_t - (I_D \otimes FC)\, h_t\right) \Big\} \\ = & \left[\lambda + \mu N + E_t + (\alpha \odot \alpha)\right]^{-1}\left[(\mu v_t + \varepsilon)\, N + E_t h_{t-1}\right] \end{aligned} \tag{3}$$

where $v_t = (I_D \otimes C^{\mathrm{T}} F^{\mathrm{H}})\hat{v}_t$ and $\varepsilon = \left(I_D \otimes C^{\mathrm{T}} F^{\mathrm{H}}\right)\hat{\varepsilon}$, which can be broken into D independent $C^{\mathrm{T}} F^{\mathrm{H}}$ transforms in realization.

$$\begin{aligned} \hat{v}_t^* = \underset{\hat{v}_t}{\operatorname{argmin}} \Big\{ & \frac{1}{2N}\left\|\hat{y} - \hat{X}_t \hat{v}_t\right\|_2^2 + \frac{\gamma}{2N}\left\|\hat{\xi}_t \hat{v}_t\right\|_2^2 + \\ & \frac{\mu}{2}\left\|\hat{v}_t - (I_D \otimes FC)\, h_t\right\|_2^2 + \hat{\varepsilon}^{\mathrm{H}}\left[(\hat{v}_t - (I_D \otimes FC)\, h_t)\right] \Big\} \end{aligned} \tag{4}$$

where $\hat{X}_t = \left[\operatorname{diag}\left(\hat{x}_t^1\right)^{\mathrm{H}}, \cdots, \operatorname{diag}\left(\hat{x}_t^{\mathrm{D}}\right)^{\mathrm{H}}\right]$, $\hat{\xi}_t = \left[\operatorname{diag}\left(\hat{\xi}_t^1\right)^{\mathrm{H}}, \cdots, \operatorname{diag}\left(\hat{\xi}_t^{\mathrm{D}}\right)^{\mathrm{H}}\right]$.

Solving Eq.(4) directly is very difficult. Fortunately, we find that $\hat{v}_t^*$ can be identically expressed as N smaller, independent objectives, over $n = [1, \cdots, N]$. For clarity of description, we define $\hat{x}_t$ and $\hat{\xi}_t$ as $\hat{x}_0$ and $\hat{x}_1$, respectively. Then, the closed-form solution of $\hat{v}_t^*(n)$ can be accelerated through Sherman-Morrison fomula [4]:

$$\begin{aligned} \hat{v}_t^*(n) = & \frac{1}{N\mu}\left[\hat{y}(n)\hat{x}_t(n) + \mu N \hat{h}_t(n) - N\hat{\varepsilon}(n)\right] \\ & - \frac{\sum_{k=0}^{1} p_k \hat{x}_k(n)}{\mu\delta}\left[\frac{1}{N}\varpi\hat{y}(n) - \sum_{k=0}^{1}\hat{x}_k^{\mathrm{H}}(n)\hat{\varepsilon}(n) \right. \\ & \left. + \mu\sum_{k=0}^{1}\hat{x}_k^{\mathrm{H}}(n)\hat{h}_t(n)\right] \end{aligned} \tag{5}$$

where $p_0 = 1$, $p_1 = \gamma$, $\delta = \mu N + \sum_{k=0}^{1} p_k \hat{x}_k^{\mathrm{H}}(n)\hat{x}_k(n)$ and $\varpi = \sum_{k=0}^{1} p_k \hat{x}_k^{\mathrm{H}}(n)\hat{x}_t(n)$.

Solution to subproblem E_t^*:

$$\begin{aligned} E_t^* &= \underset{E_t}{\operatorname{argmin}}\left\{\frac{E_t}{2}\left\|h_t - h_{t-1}\right\|_2^2 + \frac{1}{2}\left\|E_t - e\right\|_2^2\right\} \\ &= e - \frac{\left\|h_t - h_{t-1}\right\|_2^2}{2} \end{aligned} \tag{6}$$

Update the Lagrangian parameter $\hat{\varepsilon}$:

$$\hat{\varepsilon}_t^{(i+1)} = \hat{\varepsilon}_t^{(i)} + \mu\left(\hat{v}_t^{*(i+1)} - \hat{h}_t^{*(i+1)}\right) \tag{7}$$

where i and $i+1$ denotes the iteration index, the step size regularization constant μ (initially equals to 1) takes the form of $\mu^{i+1} = \min\left(\mu_{\max}, \beta\mu^i\right)$, and $\beta = 10$, $\mu_{\max} = 10000$. $\hat{h}_t^{*(i+1)} = \left(I_D \otimes C^{\mathrm{T}} F^{\mathrm{H}}\right) h_t^{*(i+1)}$, $\hat{v}_t^{*(i+1)}$ and $h_t^{*(i+1)}$ are the current solutions to the two subproblems at iteration $i+1$ within ADMM. Similar to other CF trackers [4, 6], we utilize an online adaptation strategy to improve our robustness to pose, scale and illumination changes. The appearance model is adapted online as follows:

$$\hat{x}_t^{\text{model}} = (1-\varphi)\hat{x}_{t-1}^{\text{model}} + \varphi\hat{x}_{t-1} \tag{8}$$

where $\hat{x}_t^{\text{model}}$ and $\hat{x}_{t-1}^{\text{model}}$ denote the model in the current and last frames, respectively. $\hat{x}_{t-1}$ is the training sample of the current frame. φ represents the online learning rate. Based on this strategy, we use $\hat{x}_t^{\text{model}}$ instead of $\hat{x}_{t-1}$ in Eq.(5) to compute $\hat{v}_t^*(\mathrm{n})$.

The trained filter is utilized to ascertain the new position of the target in the $t+1$ frame by the following method:

$$\phi_{t+1} = \mathcal{F}^{-1}\left(\sum_{d=0}^{1}\left(conj\left(\hat{z}_{t+1}^{d}\right)\odot\hat{v}_t^{\mathrm{d}}\right)\right) \tag{9}$$

where $\hat{z}_{t+1}^{d}$ denotes the Fourier transform of extracted feature in the $t+1$ frame. $\hat{v}_t^{\mathrm{d}}$ is the filter trained in t frame. $\mathcal{F}^{-1}$ denotes the inverse discrete Fourier transform.

Algorithm 1 FRATCF Tracker

Require: The state of object in the first frame, the following video frames.
Ensure: The estimated state in $t>1$ frame.

```
for t = 1 to end do
  if t > 1 then
    Extract feature maps from the searching area.
    Generate the response map by Eq.(9).
    Find the new location of the target.
    if t > 2 then
      Update the spatio-temporal regularization parameters
    end if
  end if
  Update the target appearance model by Eq.(8)
  Find the new location of the target.
  Train the filter with the feature residual by Eq.(3)(5)(7).
  if t > 2 then
    Update the temporal regularization parameter by Eq.(6)
  end if
end for
```

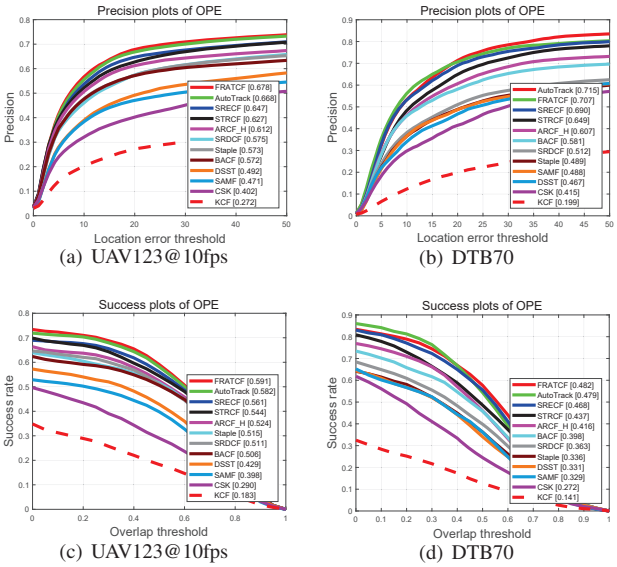
3. EXPERIMENT

3.1. Experiment setup

The experiments are conducted on two challenging UAV benchmarks, *i.e.*, UAV123@10fps[16] and DTB70[17]. We compare the proposed tracker with other 11 SOTA handcrafted-based trackers. These trackers include KCF[11], CSK[18], SAMF[19], DSST[20], ARCF[6], AutoTrack[13], Staple[21], SRECF[7], SRDCF[5], STRCF[12], BACF[4]. For parameters related to the BACF[4] and AutoTrack[13] frameworks, we set $\vartheta = 0.2$, $\tau = 2\times10^{-5}$, $\sigma = 13$, the threshold of ϱ is 3000. We set the learning rate of φ is 0.0199. The sensitivity factor of γ is chosen as 0.2. We use two ADMM iterations to train the filter quickly. In this work, DSST[20] is adopted as scale strategy for robust scale estimation. One more filter is trained in the scale space after training the translation filter. Such method has comparable performance and higher processing speed than the original one of BACF[4]. The handcrafted features including color names (CN) [22], histograms of gradient (HOG)[23], and gray-scale are fused as feature representation and appearance model for the detection and training phase. The experiments of tracking performance evaluation are conducted using MATLAB R2017b on a PC with an i5-7500 processor(3.4GHz), 16GB RAM.

Table 1: Comparison with other 11 SOTA trackers on UAV123@10fps[16] benchmark. Red, green, and blue represent the top three trackers in terms of PS and AUC, respectively.

Tracker	Venue	Prec	Succ	Tracker	Venue	Prec	Succ
FRATCF	——	0.678	0.591	Staple[21]	$16CVPR$	0.573	0.515
AutoTrack[13]	$20CVPR$	0.668	0.582	BACF[4]	$17ICVV$	0.572	0.506
SRECF[7]	$21TMM$	0.647	0.5611	DSST[20]	$21TMM$	0.492	0.429
STRCF[12]	$18CVPR$	0.627	0.544	SAMF[19]	$14ECCV$	0.471	0.398
ARCF_H[6]	$19ICCV$	0.612	0.524	CSK[18]	$12ECCV$	0.402	0.209
SRDCF[7]	$15ICVV$	0.575	0.511	KCF[11]	$15TPAMI$	0.272	0.183



(a) UAV123@10fps (b) DTB70

(c) UAV123@10fps (d) DTB70

Fig. 3: Precision plots and success plots of the proposed FRATCF and other 11 handcrafted feature based state-of-the-art trackers on UAV123@10fps[16] and DTB70[17]. The plots show the performance of these evaluated trackers under different thresholds intuitively. Notably, FRATCF ranks first in most cases in terms of PS and AUC.

Table 2: Comparison with other 11 SOTA trackers on DTB70[17] benchmark. Red, green, and blue represent the top three trackers in terms of PS and AUC, respectively.

Tracker	Venue	Prec	Succ	Tracker	Venue	Prec	Succ
FRATCF	——-	0.707	0.482	Staple[21]	16$CVPR$	0.512	0.363
AutoTrack[13]	20$CVPR$	0.715	0.479	BACF[21]	17$ICCV$	0.489	0.336
SRECF[7]	21TMM	0.690	0.468	DSST[20]	16$TPAMI$	0.467	0.331
STRCF[12]	18$CVPR$	0.649	0.437	SAMF[19]	14$ECCV$	0.488	0.329
ARCF_H[6]	19$ICCV$	0.607	0.416	CSK[18]	12$ECCV$	0.415	0.272
SRDCF[7]	17$ICVV$	0.581	0.398	KCF[11]	15$TPAMI$	0.199	0.141

Table 3: Experimental results of running the source code on our extremely cheap CPU, comparison with other 4 SOTA trackers on two benchmarks. Red, green, and blue represent the top three trackers in terms of FPS.

	FRATCF	SRECF[7]	AutoTrack[13]	BACF[4]	STRCF[12]
UAV123@10fps[16]	41.67	47.77	24.52	40.51	20.90
DTB70[17]	43.80	44.19	23.47	35.27	20.14
Venue	this work	21TMM	20$CVPR$	17$ICVV$	18$CVPR$

3.2. Comparison With SOTA Trackers

Quantitative Evaluation. For UAV123@10fps[16] and DTB70[17], we evaluate the performance of all trackers based on the one pass evaluation (OPE). Two metrics, $i.e.$, area under the curve (AUC) and precision score(PS), are employed for ranking all trackers. Besides, frame per second (FPS) is for measuring the tracking speed.

Overall Analysis. The proposed FRATCF is compared with 11 SOTA trackers using handcrafted features on the two benchmarks successively. The precision plots (PP) and success plots (SP) are exhibited in Fig.3. Meanwile, detailed results are listed in Table 1 and Table 2 for elaborate analysis. FRATCF achieved the best AUC score on all benchmarks. In terms of PS, we win the first and second on UAV123@10fps and DTB70 respectively. We can get some information from the average performance of all trackers on the two benchmarks. Although FRATCF compares with the most advanced trackers recently ($i.e.$, SRECF[7], AutoTrack[13] and ARCF[6]), it still ranks first in terms of PS and AUC. FRATCF makes an improvement of 20.1% and 18.8% in PS and AUC than the baseline BACF[4]. As can be seen from Table 3, the running speed of FRATCF is $\sim$43 FPS, which is comparable to the SRECF, which is about twice the speed of AutoTrack[13].

Attribute-based Analysis. First, Fig.4 provides the success plots under different attributes (background clutter, similar objects around and fast camera motion) from two benchmarks. We can intuitively see that FRATCF far exceeds other algorithms. Then, Table 4 shows the scores of the twelve attributes of the two benchmarks in terms of PS. According to Fig.4 and Table 4, by comparing with AutoTrack[13], we can summarize some information: In UAV123@10fps benchmark, FRATCF makes an improvement of 1.48% and 3.99% in PS and AUC in attributes of similar objects around. FRATCF outperforms AutoTrack[13] by 2.78% in PS in attributes of background clutter. In DTB70 benchmark, FRATCF performances on PS and AUC are elevated by 24% and 26.39% in attributes of background clutter. In the attributes of fast camera motion, FRATCF has a superiority of 8.89% and 10% in PS and AUC. FRATCF makes an improvement of 4% and 8.09% in PS and AUC in attributes of similar objects around. In general, it can be seen that FRATCF improves the performance in attributes of background clutter, similar objects around and fast camera motion.

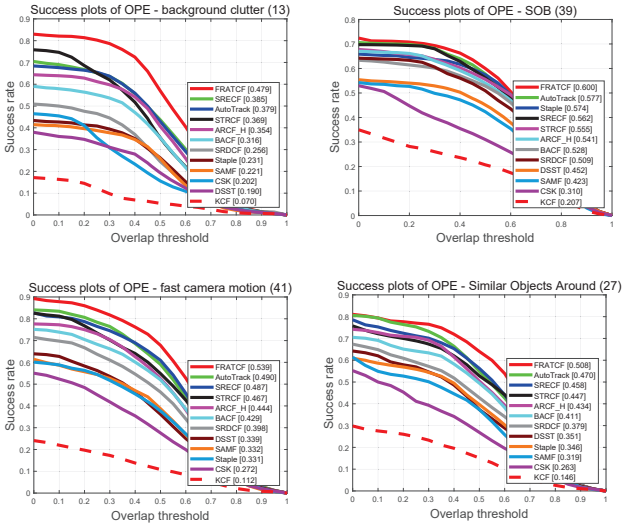


Fig. 4: Success plots of FRATCF and other handcrafted-based trackers on other attributes. FRATCF ranks first on these attributes and performs considerable improvement compared with the AutoTrack[13] and the baseline BACF[4], which proves the effectiveness and generality of the proposed method.

3.3. Qualitative Evaluation

We visualize the results of our tracker FRATCF (red rectangle), the other three SOTA trackers (SRECF, AutoTrack, ARCF_H) and the baseline BACF tracker on the sequences of Car2 and ChasingDrones. As shown in Fig.5, In the Car2 sequence, the fast camera motion became the main factor for tracking failure. Among indirect factors, the appearance of similar objects also increases the difficulty. Because the fast camera motion will cause the low resolution of the image and the appearance of the surrounding object to change. In the ChasingDrones sequence, background clutter and the fast camera motion cause tracking drift or even failure. However, only FRATCF (red rectangle) has overcome these challenging interferences and successfully tracked the target.

Table 4: Attribute based comparison of the 12 evaluated trackers on two benchmarks. Only PS is used for analysis. Moreover, red, green, and blue represent the first, second and third place respectively.

Tracker	UAV123@10fps[16]						DTB70[17]					
	ARC	LR	BC	POC	SOB	CM	OC	FCM	OV	BC	SOA	MB
Autotrack[13]	0.584	0.523	0.503	0.584	0.677	0.641	0.624	0.731	0.623	0.600	0.725	0.685
SRECF[7]	0.570	0.472	0.460	0.573	0.675	0.606	0.575	0.722	0.694	0.616	0.706	0.664
STRCF[12]	0.524	0.509	0.477	0.559	0.630	0.602	0.617	0.713	0.652	0.611	0.677	0.689
ARCF_H[6]	0.522	0.483	0.428	0.531	0.657	0.551	0.546	0.654	0.671	0.555	0.679	0.590
SRDCF[5]	0.472	0.431	0.389	0.504	0.585	0.527	0.478	0.554	0.552	0.393	0.569	0.664
Staple[21]	0.459	0.408	0.409	0.507	0.612	0.499	0.528	0.494	0.420	0.393	0.529	0.332
BACF[4]	0.478	0.431	0.425	0.467	0.605	0.532	0.515	0.622	0.567	0.499	0.624	0.617
DSST[20]	0.386	0.364	0.341	0.424	0.517	0.411	0.472	0.482	0.493	0.287	0.542	0.346
SAMF[19]	0400	03.44	0.278	0.421	0.508	0.389	0.500	0.499	0.531	0.352	0.500	0.345
CSK[18]	0.294	0.309	0.227	0.329	0.432	0.306	0.397	0.443	0.471	0.283	0.433	0.333
KCF[11]	0.229	0.210	0.098	0.212	0.315	0.180	0.219	0.164	0.173	0.070	0.222	0.121
FRATCF	0.606	0.548	0.517	0.578	0.687	0.637	0.663	0.796	0.703	0.744	0.754	0.790

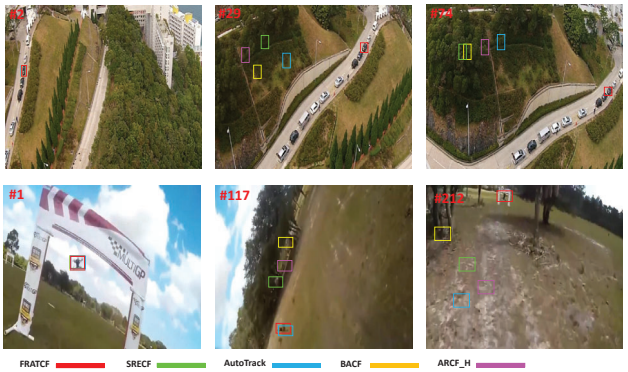


Fig. 5: Qualitative evaluation of the proposed FRATCF, top three trackers on average performance and the baseline BACF.

4. CONCLUSION

In this article, we propose a feature residual filter with automatic spatio-temporal regularization. The aim is to further strengthen the training of the filter by learning the feature residuals between two adjacent frames in the training phase, obtain better accuracy and improve the performance in background clutter, similar objects around and fast-moving camera scenes. Experimental results on two challenging benchmarks show that the proposed tracker FRATCF has certain advantages in terms of accuracy and running speed, which is sufficient for real-time UAV applications.

References

[1] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, and L. A. Saidane, "Monitoring road traffic with a uav-based system," in *WCNC*, pp. 1–6, IEEE, 2018.

[2] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments," in *IROS*, pp. 229–236, IEEE, 2019.

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

[4] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, pp. 1135–1143, 2017.

[5] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *CVPR*, pp. 4310–4318, 2015.

[6] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *ICCV*, pp. 2891–2900, 2019.

[7] C. Fu, J. Jin, F. Ding, Y. Li, and G. Lu, "Spatial reliability enhanced correlation filter: An efficient approach for real-time uav tracking," *IEEE Trans. Multimedia*, 2021.

[8] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *CVPR*, pp. 4670–4679, 2019.

[9] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *CVPR*, pp. 4282–4291, 2019.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, pp. 2544–2550, IEEE, 2010.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2014.

[12] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *CVPR*, pp. 4904–4913, 2018.

[13] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization," in *CVPR*, pp. 11923–11932, 2020.

[14] F. Zhang, S. Ma, Y. Zhang, and Z. Qiu, "Perceiving temporal environment for correlation filters in real-time uav tracking," *IEEE Signal Process. Lett.*, 2021.

[15] S. Li, Y. Liu, Q. Zhao, and Z. Feng, "Learning residue-aware correlation filters and refining scale estimates with the grabcut for real-time uav tracking," *arXiv preprint arXiv:2104.03114*, 2021.

[16] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *ECCV*, pp. 445–461, Springer, 2016.

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

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, pp. 702–715, Springer, 2012.

[19] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV*, pp. 254–265, Springer, 2014.

[20] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, 2016.

[21] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, pp. 1401–1409, 2016.

[22] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, pp. 1090–1097, 2014.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, pp. 886–893, Ieee, 2005.