Sample Purification-Aware Correlation Filters for UAV Tracking with Cooperative Deep Features

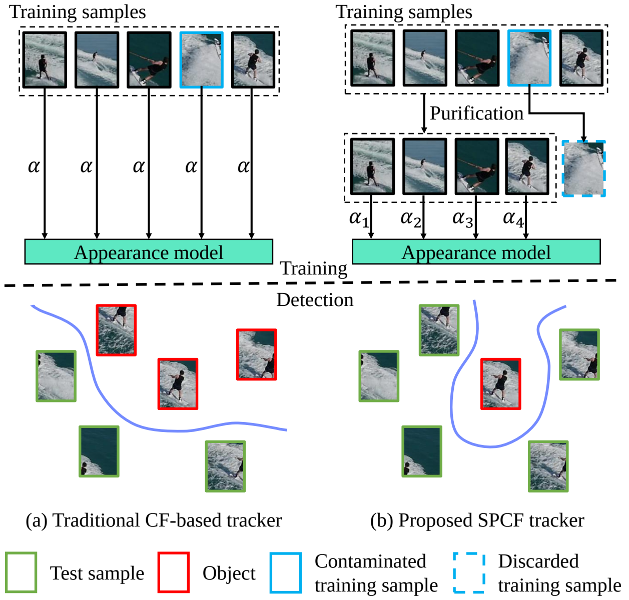
Changhong Fu[1,*], Fuling Lin[1], Fan Li[1], and Yujie He[1]

***Abstract*— Correlation Filters (CF) have recently demonstrated promising performance in terms of rapidly tracking objects for unmanned aerial vehicles (UAV) in different types of UAV tracking tasks. The strength of the approach comes from its ability to learn how the object is changing over time efficiently. However, due to heavy dependence on the quality of the training set and the lack of real negative training samples, the object appearance model may be easily interfered by the corrupted training samples, which can result in suboptimal performance. Besides, limited by the representation of a single feature, the tracking model may fail to cope with the complex surrounding environment and considerable appearance variation of the object. In this work, the principal causes behind the problems of the abundance of object representation and purity of training samples have been tackled, to simultaneously improve both discriminative power and anti-disturbance capability of the tracking model. Comprehensive experiments on 100 challenging UAV image sequences have demonstrated that the novel sample purifying tracker based on cooperative features learning, i.e., SPCF tracker, outperforms 15 state-of-the-art trackers in terms of efficiency, robustness, and accuracy. To the best of our knowledge, the presented SPCF tracker is designed for object tracking and employed in UAV tracking tasks for the first time.**

I. INTRODUCTION

Visual tracking plays an increasingly important role with the development of unmanned aerial vehicles (UAV), such as surveillance of autonomous landing [1], aerial refueling [2], crowd monitoring [3], forest fire surveillance [4] and obstacle avoidance [5]. Measurable progress has been made in tracking for UAV in recent years. However, UAV tracking still faces many problems due to many factors, including UAV motion, background clutter, similar object, illumination, and scale variation. Additionally, aerial operating environment increases difficulty in tracking such as mechanical vibration and airflow influence.

Correlation Filter (CF) has become a widely used framework for tracking problems due to its high computational efficiency and tracking performance, especially for object tracking in the field of UAVs. CF-based trackers can learn and detect the object quickly by carrying out the element-wise calculation in the frequency domain. However, traditional CF-based trackers struggle when the training set is polluted by contaminated sample. Unfortunately, the rapid motion of objects, out-of-view, and many other factors can easily lead to corruption of the sample set. In other words, in addition to samples with various perspectives of the object, there are samples without object information in the sample set, i.e., samples having blue boxes in Fig. 1. Furthermore, the conventional CF-based trackers apply only one handcrafted feature, which ignores the high-level semantic information of the convolutional feature and can not adequately and comprehensively describe the object appearance.



(a) Traditional CF-based tracker (b) Proposed SPCF tracker

Fig. 1. Comparison between traditional CF-based and proposed tracker, i.e., SPCF. Training samples are obtained in different frames and used to train an appearance model. As for the images with blue boxes, they are contaminated samples with excessive background interference and no object appearance information. Traditional CF-based tracker gives each sample the same weight, while the proposed tracker assigns different weights to the purified samples. Through discarding the contaminated samples and assigning adaptive weights, the appearance model can be more robust and discriminative to avoid model drift and loss of the object.

To deal with the above challenging problems, we focus on achieving better representations of the object with multiple deep features and purifying training samples using sample weights in this work. A novel kernelized correlator with multi-feature constraint is proposed, i.e., SPCF tracker. The main contributions of this work are listed as follows:

- A new integrated learning method for multi-convolutional features is developed. Different deep features are used to train different filters, and each filter continuously interacts with each other to repress the anomalies in different feature response, see Fig. 2.
- A novel developed joint training framework is proposed to both learn sample weights and filter, which can increase the importance of correct samples and decrease

[1] C. Fu, F. Lin, F. Li, and Y. He are with the School of Mechanical Engineering, Tongji University, 201804 Shanghai, China e-mail: changhongfu@tongji.edu.cn.

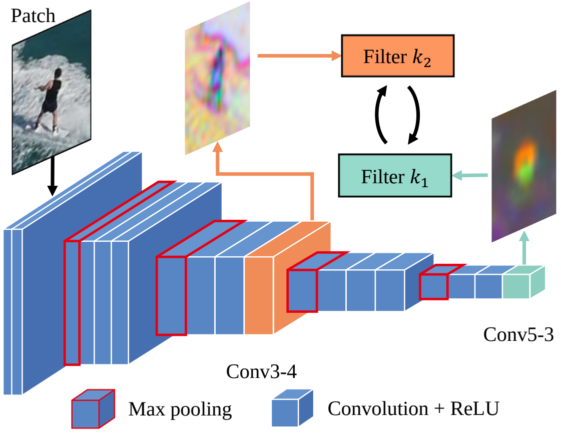


Fig. 2. Cooperative deep features. Cooperative features are extracted from the convolutional layers conv5-4 and conv3-4 in VGG-19.

the weights of corrupted ones.

- A new appearance model update strategy is proposed. The tracker purifies training samples by evaluating the quality of prior frames based on the current filter and discarding samples that obtain the high loss.
- Experiments on 100 challenging UAV image sequences demonstrate that the tracker performs favorably against other 15 state-of-the-art trackers in accuracy, robustness, and efficiency.

To the best of our knowledge, the presented SPCF tracker is designed for object tracking and employed in UAV tracking tasks for the first time.

The rest of this paper is organized as follows. Section II describes related works. The elements of the proposed SPCF tracker are introduced in Section III. Experimental results and comparisons with other state-of-the-art trackers are proposed in Section IV. Finally, conclusions are presented in Section V.

II. RELATED WORKS

A. *Correlation filters for tracking*

The CF-based trackers have attracted full attention in the field of object tracking due to their high computational efficiency. Many trackers have applied CF-based framework, including the minimum output sum of squared error [6], kernelized correlation filters [7], discriminative scale space tracking [8], hierarchical convolutional features [9], co-trained kernelized correlation filters [10], and many other trackers [11]–[14]. The base sample implicitly extends the training sample set that is used to train a CF-based tracker. Therefore, the quality of the training set directly determines the outcome of the tracking model. However, traditional CF trackers have a natural defect when the training set is polluted by contaminated sample.

B. *Multiple representations for tracking*

In traditional CF-based tracker, the appearance model is trained continuously using one feature, i.e., histograms of gradients (HOG) [15], color names (CN) [16], etc. It is challenging when the object experiences deformation or illumination in the course of tracking. Only one feature used to describe the object is insufficient to construct a discriminative appearance model. In [11], [12], [17], multiple handcrafted features (CN and HOG) are combined to achieve robustness to deformation and illumination variation so that the weaknesses of features can be reciprocally compensated. Additionally, features extracted from convolutional neural networks (CNNs) are also applied in visual tracking, which carries the semantic information of the object and can significantly improve the tracking performance [18]–[20]. Deep features from different CNN layers are utilized for the object pyramid representation so that the appearance model can be better encoded [9]. Multiple representations can improve the discriminative power of the appearance model, but the practical evaluation of samples is not considered, which makes the model susceptible to interference of negative samples.

C. *Approaches to the problem of corrupted samples*

When corrupted samples contaminate the training set, traditional tracking-by-detection methods struggle easily. Many factors will lead to those corrupted samples, such as temporary object drift and background interference. When not accurate enough tracking prediction leads to the object drift temporarily, or the positive training samples contain too much background information, the search area will contain little object information, i.e., those samples are contaminated. Efforts have been made to investigate the problem of sample contamination. MOOSE [6] manages training set by directly discarding samples that do not meet a specific benchmark. TGPR [21] uses distance comparisons to manage the training set, and MEEM [22] uses a combination of experts. SRDCFdecon [23] evaluates the quality of the samples to manage the training set dynamically. Managing sample set can improve the quality of the training set, but those trackers commonly misjudge the uncontaminated samples as contaminated samples due to using only a kind of feature.

III. PROPOSED TRACKING APPROACH

In this section, kernelized correlation filters, i.e., correlators, with purified sample and cooperative features learning, is proposed. Its main structure is shown in Fig. 3.

A. *Cooperative kernelized correlator*

The proposed tracker is constructed based on kernel trick [14]. For a non-linear feature mapping function φ, the kernel trick can calculate the inner product between the unlabeled input $\mathbf{z}_i$ and each of the training inputs $\mathbf{x}$ efficiently. Thus, the Eq. (1) can be developed as $\kappa : \mathbb{R}^M \times \mathbb{R}^M \mapsto \mathbb{R}$.

$$\kappa\left(\mathbf{x},\mathbf{z}_i\right)=\varphi(\mathbf{x})^\top\varphi\left(\mathbf{z}_i\right)\ , \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^M$ denotes the vectorized feature of tracking object region. The sample-based vectors, $\mathbf{z}_i\in\mathcal{T}(\mathbf{z})\in\mathbb{R}^M$, is generated from the test sample $\mathbf{z}$ by the predefined transform function $\mathcal{T}(\cdot)$. The kernel vector $\mathbf{k}^{\mathbf{xz}} = [k_1^{\mathbf{xz}},\cdots,k_n^{\mathbf{xz}}]^\top$

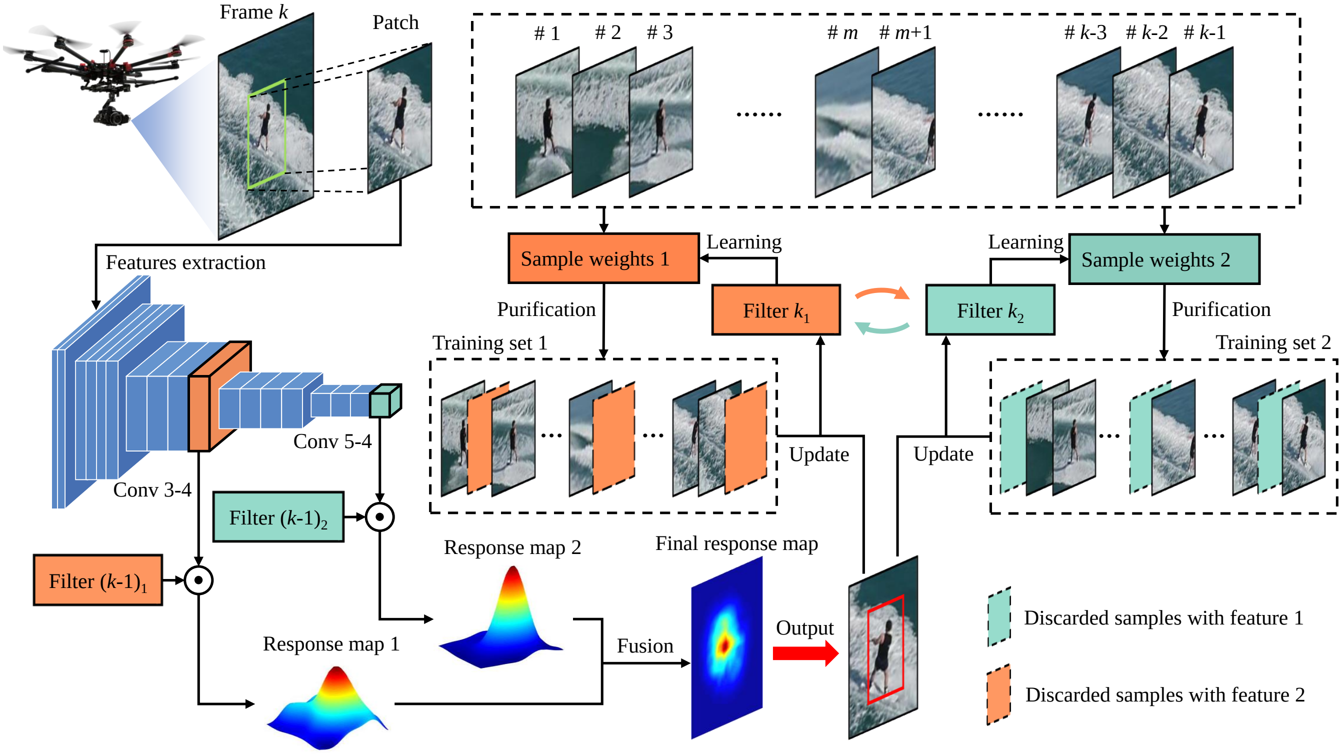


Fig. 3. The main workflow of the presented tracking approach, i.e., SPCF tracker. Given the image in the frame k, the patch is obtained based on the predicted object position in the frame $k-1$. The patch is encoded by two deep features extracted from the different layers of VGG-19 and then mapped to non-linear space using kernel trick. Final response map is obtained by fusing response maps. Especially, filters are trained using purified training sets, which are obtained by discarding contaminated samples from samples in previous frames. By searching for the highest value in the final response map, the tracking result in the frame k can be estimated.

is a linear combination of a set of sample-based vectors $k_i^{\mathbf{xz}} = \kappa\left(\mathbf{x}, \mathbf{z}_i\right)$.

Therefore, the kernel correlator can be denoted by

$$C(\mathbf{x}, \mathbf{z}) = \mathbf{w} \star \mathbf{k}^{\mathbf{xz}} \ , \tag{2}$$

where $\star$ is the correlation operstor, and $\mathbf{w}$ denotes the learned correlation filter. As a result, Eq. (2) can be expressed in the frequency domain as:

$$\hat{C}(\mathbf{x}, \mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\mathbf{w}}^{*} \ , \tag{3}$$

where $\odot$ indicates the element-wise product. $\hat{}$ denotes the discrete Fourier transform (DFT) of a signal, and * denots complex conjugate.

For single feature case, the proposed tracker can be constructed in the frequency domain by minimizing the cost function $\hat{\mathcal{L}}(\mathbf{w})$ as follows:

$$\hat{\mathcal{L}}(\mathbf{w}) = \left\| \hat{C}\left(\mathbf{x}, \mathbf{x}\right) - \hat{\mathbf{y}} \right\|_2^2 + \lambda \hat{\mathbf{k}}^{\mathrm{xx}} \left\| \hat{\mathbf{w}}^{*} \right\|_2^2 \ . \tag{4}$$

For multi-features case, considering the correlation between different features, the optimal correlation filter in the frequency domain $\hat{\mathbf{w}}^{*}$ can be obtained by minimizing the objective:

$$\begin{aligned} \hat{\mathcal{L}}(\mathbf{w}_n^{*}) = &\sum_{n=1}^{N} \left(\left\| \hat{C}_n\left(\mathbf{x}_n, \mathbf{x}_n\right) - \hat{\mathbf{y}}_n \right\|_2^2 \right. \\ &\left. + \lambda_n \hat{\mathbf{k}}^{\mathbf{x}_n\mathbf{x}_n} \left\| \hat{\mathbf{w}}_n^{*} \right\|_2^2 \right) \\ &+ \gamma \sum_{i,j=1}^{N} \left\| \hat{C}_i\left(\mathbf{x}_i, \mathbf{x}_i\right) - \hat{C}_j\left(\mathbf{x}_j, \mathbf{x}_j\right) \right\|_2^2 \end{aligned} \ , \tag{5}$$

where N is the numbers of features. $\mathbf{y}_n \in \mathbb{R}^M$ is the desired correlation output with the training example $\mathbf{x}_n$. λ_n is a regularization parameter that controls overfitting effect. γ is the penalty factor which is used to control the correlation output of different features.

Remark 1: The first term in the Eq. (5) denotes the loss of n-th feature. The second term is a regularization term. The third term connotes the constraint relationship between different features to repress the anomalies in different feature response.

Given that the filter should be similar to the previous one, the learning methods should be passive. Besides, the learning methods should be aggressive enough to classify the test samples correctly. Thus, a novel method is proposed to update the model with the previous best samples adaptively.

The joint loss $\hat{\mathcal{J}}$ is introduced:

$$\begin{aligned}\hat{\mathcal{J}}(\alpha_n^s, \mathbf{w}_n^*) =& \sum_{n=1}^{N}\sum_{s=1}^{S}\alpha_n^s\Bigg(\lambda_n\hat{\mathbf{k}}^{\mathbf{x}_n^s\mathbf{x}_n^s}\left\|\hat{\mathbf{w}}_n^*\right\|_2^2 \\ &+ \left\|\hat{C}_n^s\left(\mathbf{x}_n^s, \mathbf{x}_n^s\right) - \hat{\mathbf{y}}_n\right\|_2^2\Bigg) \\ &+ \gamma\sum_{i,j=1}^{N}\sum_{s=1}^{S}\left\|\hat{C}_i^s\left(\mathbf{x}_i^s, \mathbf{x}_i^s\right) - \hat{C}_j^s\left(\mathbf{x}_j^s, \mathbf{x}_j^s\right)\right\|_2^2 , \\ &+ \sum_{n=1}^{N}\left(\mu_n\sum_{s=1}^{S}\frac{(\alpha_n^s)^2}{t^s}\right)\end{aligned} \tag{6}$$

where $\alpha_n^s \in \mathbb{R}$ and t^s are the weights of the s-th selected frame with n-th feature and temporal weights, respectively. S is the number of selected frames, at which samples are selected for training. μ_n is a regularization parameter.

Remark 2: The standard weighted loss (4) is developed, as shown in Eq. (6). The function is constituted by both correlation filter $\mathbf{w}_n$ and the sample weights α_n^s. As a result, the weights α_n^s are no longer constant.

B. Optimization

The problem is how to optimize Eq. (6) for learning parameter $\mathbf{w}_n^*$ and α_n^s. The two parameters can be optimized alternatively, which means that the ~~setting~~ $\mathbf{w}_n^*$ is trained with fixed α_n^s firstly, and then α_n^s is trained with fixed $\mathbf{w}_n^*$. Each of the subproblems, $\mathbf{w}_n^*$ and α_n^s, have closed-form solutions.

Subproblem $\hat{\mathbf{w}}_n^*$:

In this section, the value of N is assumed as 2. Therefore, the optimization problem (6) can be solved by setting the first derivative of $\hat{\mathbf{w}}_1^*$ and $\hat{\mathbf{w}}_2^*$ to zero, i.e.:

$$\frac{\partial\hat{\mathcal{J}}}{\partial\hat{\mathbf{w}}_1^*} = 0,\ \frac{\partial\hat{\mathcal{J}}}{\partial\hat{\mathbf{w}}_2^*} = 0 , \tag{7}$$

Since all the operations in (7) are performed in element wise, the elements of $\hat{\mathbf{w}}_1^*$ and $\hat{\mathbf{w}}_2^*$ can be solved independently as follows:

$$\begin{aligned}\hat{\mathbf{w}}_1^* &= \frac{\mathbf{A}_{22}^*\odot\mathbf{A}_1^*\odot\hat{\mathbf{y}}_1^* + \gamma\mathbf{A}_2^*\odot\hat{\mathbf{y}}_2\odot\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s}}{\mathbf{A}_{11}^*\odot\mathbf{A}_{22}^* - \gamma^2\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s}\odot\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s}} \\ \hat{\mathbf{w}}_2^* &= \frac{\mathbf{A}_{11}^*\odot\mathbf{A}_2^*\odot\hat{\mathbf{y}}_2^* + \gamma\mathbf{A}_1^*\odot\hat{\mathbf{y}}_1\odot\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s}}{\mathbf{A}_{22}^*\odot\mathbf{A}_{11}^* - \gamma^2\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s}\odot\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s}}\end{aligned} , \tag{8}$$

where the operator denotes element-wise product and $\mathbf{A}_{11}, \mathbf{A}_{22}$ are computed as follows:

$$\begin{aligned}\mathbf{A}_{11} &= \sum_{s=1}^{S}\alpha_1^s\left(\frac{(\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s})^*\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s}}{\hat{\mathbf{k}}^{\mathbf{x}_2^S\mathbf{x}_2^S}}\right) + \lambda_1 + \gamma(\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s})^* \\ \mathbf{A}_{22} &= \sum_{s=1}^{S}\alpha_2^s\left(\frac{(\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s})^*\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s}}{\hat{\mathbf{k}}^{\mathbf{x}_1^S\mathbf{x}_1^S}}\right) + \lambda_2 + \gamma(\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s})^*\end{aligned} . \tag{9}$$

As for $\mathbf{A}_1$ and $\mathbf{A}_2$, they can be obtained by:

$$\begin{aligned}\mathbf{A}_1 &= \sum_{s=1}^{S}\alpha_2^s\left(\frac{\hat{\mathbf{k}}^{\mathbf{x}_1^s\mathbf{x}_1^s}}{\hat{\mathbf{k}}^{\mathbf{x}_1^S\mathbf{x}_1^S}}\right) \\ \mathbf{A}_2 &= \sum_{s=1}^{S}\alpha_1^s\left(\frac{\hat{\mathbf{k}}^{\mathbf{x}_2^s\mathbf{x}_2^s}}{\hat{\mathbf{k}}^{\mathbf{x}_2^S\mathbf{x}_2^S}}\right)\end{aligned} . \tag{10}$$

Subproblem α_n^s:

The second step of optimization is to train α_1^s and α_2^s with fixed $\hat{\mathbf{w}}_1^*$ and $\hat{\mathbf{w}}_2^*$, which requires setting the first derivative of α_1^s and α_2^s to zero, respectively. Therefore, α_1^s can be obtained by:

$$\begin{aligned}\frac{\partial\hat{\mathcal{J}}}{\partial\alpha_1^s} =& \frac{\partial}{\partial\alpha_1^s}\Bigg(\sum_{s=1}^{S}\alpha_1^s\left\|\hat{C}_1\left(\mathbf{x}_1^s, \mathbf{x}_1^s\right) - \hat{\mathbf{y}}_1\right\|_2^2 \\ &+ \mu_1\sum_{s=1}^{S}\frac{(\alpha_1^s)^2}{t^s}\Bigg) = 0\end{aligned} . \tag{11}$$

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

$$\begin{aligned}\min \quad & \hat{\mathcal{J}} = \sum_{s=1}^{S}(\beta_1^s\alpha_1^s + \gamma_1^s(\alpha_1^s)^2) \\ \text{s.t.} \quad & \sum_{s=1}^{S}\alpha_1^s = 1\end{aligned} . \tag{12}$$

where β_1^s and γ_1^s are set to be $\|\hat{C}_1\left(\mathbf{x}_1^s, \mathbf{x}_1^s\right) - \hat{\mathbf{y}}_1\|_2^2$ and μ_1/t^s. Thus, α_1^s can be solved using Lagrangian method:

$$\begin{aligned}L(\alpha_1^s, \lambda, \lambda_1^s) =& \sum_{s=1}^{S}\left(\beta_1^s\alpha_1^s + \gamma_1^s\left(\alpha_1^s\right)^2\right) \\ &+ \lambda\left(\sum_{s=1}^{S}\alpha_1^s - 1\right) - \sum_{s=1}^{S}\lambda_1^s\alpha_1^s = 0 , \\ \text{s.t.}\ \ & \beta_1^s + 2\gamma_1^s\alpha_1^s + \lambda - \lambda_1^s = 0 , \\ & \lambda_1^s\alpha_1^s = 0 , \\ & \sum_{s=1}^{S}\alpha_1^s - 1 = 0 , \\ & \alpha_1^s \geq 0 , \\ & \lambda_1^s \geq 0 .\end{aligned} \tag{13}$$

α_2^s can be derived in the same way.

C. Purification of samples

Due to heavy dependence on the quality of the training set and the lack of real negative training samples, traditional CF trackers easily struggle when the training set is polluted by contaminated sample. Many factors, such as the rapid motion of objects, out-of-view, and background interference, can lead to corruption of samples.

In this work, the current filter is used to calculate the responses with prior samples to evaluate the quality of the samples in selected frames. On the one hand, the weights of samples are based on that quality to increase the importance of correct samples and decrease the weights of corrupted ones. On the other hand, when the number of samples exceeds the pre-setting sample size $S_{\max}$, the sample that obtains the lowest sample weight will be discarded, as shown in Fig. 4. As for prior information, it is absorbed using the temporal weights t^s in Eq. (6). Considering that closer samples, i.e., the s_0 most recent samples, usually are more reliable t^s is designed as follows:

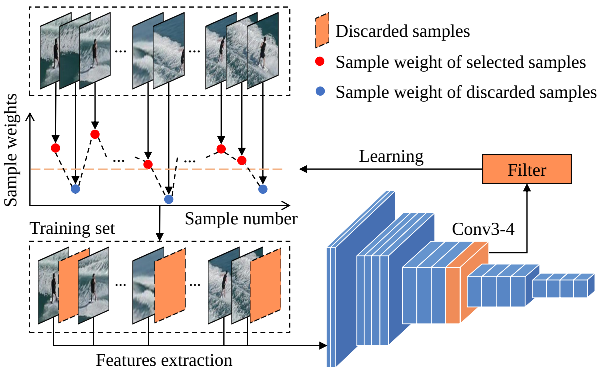


Fig. 4. Sample purification. The current filter is used to calculate the responses with prior samples to evaluate the quality of the samples in selected frames. Sample weights are obtained based on that quality. When the number of samples exceeds the pre-setting sample size $S_{\max}$, the sample that obtains the lowest sample weight will be discarded.

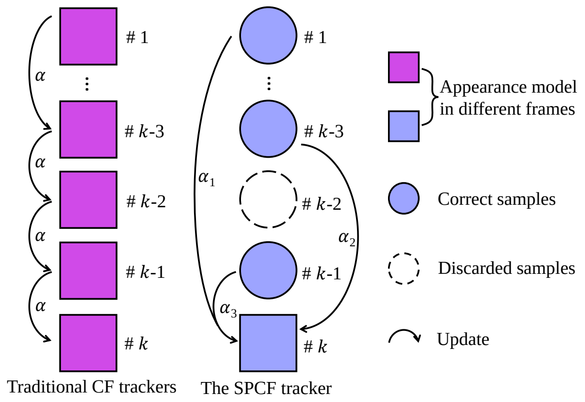


Fig. 5. Update mechanism of traditional CF-based trackers and the proposed SPCF tracker.

$$t^s = \begin{cases} a, & s = 1, ..., S - s_0 \\ a(1-q)^{S-s_0-s}, & s = S - s_0 + 1, ..., S \end{cases}, \quad (14)$$

where the constant $a = (S - s_0 + \frac{(1-q)^{-s_0}-1}{q})^{-1}$ is determined by the condition $\sum_s t^s = 1$. q and s_0 are superparameters.

Remark 3: In this work, considering that in most cases, the status and location of the object will not change rapidly, the s_0 most recent samples are given larger temporal weights t^s.

D. Model update

Appearance model is updated with below formulation:

$$\mathbf{x}^s_{n(\text{model})} = \begin{cases} \sum_{s=1}^{S}(\frac{\alpha_n^s t^s}{\sum_{s=1}^{S} \alpha_n^s t^s}\mathbf{x}_n^s), & S \leq S_{\max} \\ \sum_{s=1}^{S_{\max}}(\frac{\alpha_n^s t^s}{\sum_{s=1}^{S_{\max}} \alpha_n^s t^s}\mathbf{x}_n^s), & S > S_{\max} \end{cases}, \quad (15)$$

where $\mathbf{x}^s_{n(\text{model})}$ is appearance model that is obtained from the combination of $\mathbf{x}^s_n$. α^s_n and t^s are the sample weight and temporal weight of the sample $\mathbf{x}^s_n$, respectively.

Remark 4: As shown in Fig. 5, traditional CF-based trackers usually update appearance model frame-by-frame with constant learning rate, which leads to that the training set includes both contaminated samples and correct samples. The proposed tracker obtains appearance model from samples in purified frames with different weights. The training set discards contaminated samples, and correct samples obtain higher weights.

E. Combination of convolutional features

Most CF-based trackers use only a kind of feature, which is not adequate to describe the tracked object when the object appearance suffers changing or camera moves quickly, especially in UAV tracking field. Deep convolutional features are receiving more attention in recent years for enhancing performance.

In this work, the features are extracted from the convolutional layers conv3-4 and conv5-4 to train the proposed tracker. Since deeper layers of CNN can capture more semantic information, while the shallower includes more spatial data, it is necessary to combine deep and shallow features to obtain more robust and accurate performance.

Remark 5: In this work, the layers conv3-4 and conv5-4 of VGG-Net [24] are used to extract the depp features, as shown in Fig. 2.

IV. EXPERIMENTS

In this section, the experimental performance assessment attained by evaluating the proposed SPCF tracker on 100 challenging UAV image sequences from a well-known and frequently-used UAV123 benchmark [25]. This benchmark is captured from low-altitude UAVs and includes exhaustive UAV tracking challenges, i.e., background clutter (BC), aspect ratio change (ARC), fast motion (FM), camera motion (CM), illumination variation (IV), full occlusion (FOC), out-of-view (OV), low resolution (LR), scale variation (SV), partial occlusion (POC), viewpoint change (VC), and similar object (SOB).

A. Evaluation criteria

This work uses two standard evaluation metrics to estimate the tracking performance, i.e., center location error (CLE) and success rate (SR) based on one-pass evaluation (OPE). The CLE is defined as the Euclidean distance between the center of the estimated bounding box and ground-truth position, which can measure the precision. The SR is characterized as the intersection over union (IoU) of the tracker bounding box and ground-truth bounding box, which can indicate the accuracy of the estimate of the scale.

Remark 6: This work evaluates the trackers based on protocol in the visual tracking area strictly. The ranking of precision is based on the CLE threshold, which is set to 20 pixels. The area under curve (AUC) is used to rank the trackers in success.

B. Comparison with state-of-the-art trackers

The experimental results achieved by SPCF tracker and other 15 state-of-the-art trackers are compared based on the norm mentioned above, including MCCT-H, MCCT [13], MUSTER [26], DCF_CA, SAMF_CA [27], STRCF [28],

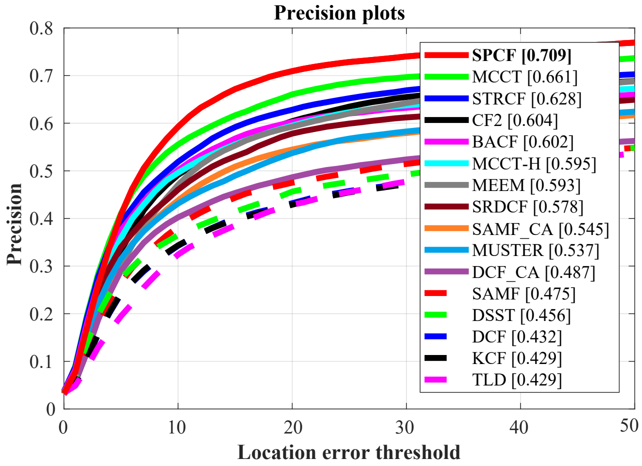


Fig. 6. PPs of 16 tracking approaches on 100 challenging aerial image sequences. SPCF tracker performs 7.3% and 12.9% better than the second (MCCT) and the third trackers (STRCF), respectively.

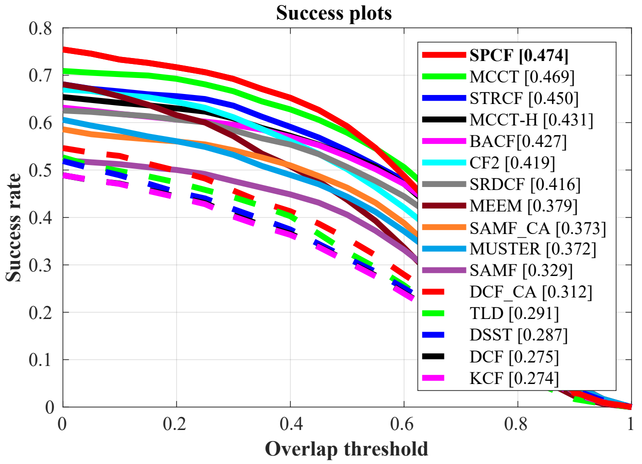


Fig. 7. SPs of 16 tracking approaches on 100 challenging aerial image sequences. SPCF tracker performs 1.1% and 5.3% better than the second (MCCT) and the third trackers (STRCF), respectively.

CF2 [9], BACF [29], SRDCF [30], CoKCF [10], SAMF [11], DCF, KCF [7], KCC [14], and DSST [8]. Fig. 6 and 7 show all the results of precision plots (PPs) and success plots (SPs) on 100 challenging UAV image sequences, respectively. As shown in Fig. 6, the SPCF outperforms the second-best tracker MCCT (2018 CVPR) and third-best tracker STRCF (2018 CVPR) by a gain of 7.2% and 12.9%, respectively. Similar to the results in SPs, SPCF is also ranking No.1 among other 15 state-of-the-art trackers in success rate, as shown in Fig. 7.

Remark 7: All trackers are implemented with MATLAB R2017a, and all the experiments are running on the computer with an i7-8700K processor (3.7GHz), 48GB RAM and NVIDIA Quadro P2000 GPU.

C. Attribute-based evaluation

Besides overall performance, the attribute-based performance of SPCF and 15 other trackers are also compared. Fig. 8 and Fig. 9 are boxplots of all evaluated trackers on different attributes of 100 challenging UAV image sequences from UAV123 [25] benchmark, respectively. As shown in Table I, the SPCF tracker favorably outperforms other state-of-the-art trackers in all aspects of precision. Table II indicates that the SPCF tracker is ranking No.1 except for BC, OV, and SOB attributes.

D. Limitations

Performance-based on attributes. Although the proposed SPCF tracker outperforms other 15 state-of-the-art trackers in precision, in terms of success rate, the SPCF tracker ranks second only to MCCT on the BC, OV and SOB attribute. The BC attribute indicates how background information clutters the tracker, and the OV attribute denotes the situation where the object leaves the field of view. As for the SOB attribute, this attribute expresses the tracking performance when there is a similar background with the object. The performance tracker can be better improved by adding more background information.

Performance on speed. The frame per second (FPS) of SPCF tracker reaches only 2.30, although the proposed SPCF tracker has outperformed recent 15 state-of-the-art tracking approaches in the experiment.

Remark 8: The results of the experiments are obtained on Matlab without optimization for speed. Thanks to the excellent load capacity of the ample storage space of UAVs, multiple GPUs can be carried to improve the calculating speed of the proposed SPCF tracker.

V. CONCLUSION

In this work, a novel cooperative features learning tracker based on sample purification, i.e., SPCF tracker, is proposed to solve challenging UAV tracking problems. Joint loss is minimized to update both sample weights and correlation filters, where sample weights are used to increase the importance of correct samples. Also, the current filters are used to evaluate the quality of prior frames to purify training samples, to simultaneously improve both discriminative power and anti-disturbance capability of the tracking model. Moreover, multiple convolutional features are incorporated into the training process of correlators to obtain accuracy and robust performance. The performance assessment on 100 challenging UAV image sequences has demonstrated that SPCF tracker achieves the best performance among other 15 state-of-the-art trackers in terms of robustness and precision. We firmly believe that the results of the proposed SPCF tracker would further improve the CF-based framework in the field of UAV tracking.

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).

TABLE I

SCORES OF PP (CLE = 20 PIXELS). RED , GREEN, AND BLUE FONTS INDICATE THE 1ST, 2ND, AND 3RD PERFORMANCES AMONG 16 TRACKERS.

	ARC	BC	CM	FM	FOC	IV	LR	OV	POC	SV	SOB	VC
STRCF	0.525	0.433	0.614	0.470	0.389	0.488	0.480	0.496	0.536	0.578	0.605	0.538
MCCT	0.573	0.459	0.635	0.465	0.423	0.570	0.499	0.529	0.591	0.614	0.692	0.583
MEEM	0.513	0.475	0.552	0.315	0.375	0.513	0.482	0.445	0.518	0.538	0.603	0.549
KCF	0.326	0.184	0.331	0.232	0.271	0.303	0.293	0.335	0.352	0.395	0.463	0.365
DCF	0.334	0.188	0.334	0.252	0.273	0.309	0.290	0.335	0.355	0.398	0.468	0.370
DCF_CA	0.341	0.294	0.408	0.260	0.281	0.347	0.339	0.344	0.365	0.426	0.480	0.417
MCCT-H	0.488	0.386	0.551	0.346	0.344	0.478	0.439	0.468	0.525	0.543	0.603	0.486
CF2	0.517	0.451	0.585	0.361	0.441	0.552	0.414	0.497	0.530	0.551	0.606	0.543
SAMF_CA	0.405	0.307	0.505	0.407	0.321	0.367	0.359	0.475	0.428	0.489	0.531	0.444
SAMF	0.391	0.202	0.387	0.353	0.305	0.320	0.307	0.431	0.405	0.449	0.508	0.401
BACF	0.501	0.419	0.583	0.439	0.333	0.466	0.427	0.460	0.498	0.549	0.593	0.539
SRDCF	0.483	0.331	0.545	0.434	0.390	0.439	0.406	0.480	0.484	0.534	0.581	0.481
MUSTER	0.444	0.321	0.503	0.339	0.390	0.385	0.448	0.425	0.444	0.495	0.540	0.472
DSST	0.354	0.184	0.352	0.266	0.287	0.318	0.334	0.381	0.384	0.431	0.506	0.380
TLD	0.379	0.252	0.396	0.275	0.270	0.289	0.420	0.295	0.341	0.405	0.479	0.382
SPCF	0.644	0.483	0.708	0.552	0.464	0.626	0.559	0.554	0.628	0.670	0.716	0.662

TABLE II

SCORES OF SP (MEASURED BY AUC). RED , GREEN, AND BLUE FONTS INDICATE THE 1ST, 2ND, AND 3RD PERFORMANCES AMONG 16 TRACKERS.

	ARC	BC	CM	FM	FOC	IV	LR	OV	POC	SV	SOB	VC
STRCF	0.361	0.277	0.448	0.301	0.195	0.344	0.265	0.350	0.355	0.410	0.436	0.393
MCCT	0.387	0.300	0.464	0.309	0.218	0.396	0.283	0.366	0.392	0.430	0.481	0.419
MEEM	0.316	0.285	0.366	0.196	0.179	0.329	0.232	0.293	0.323	0.336	0.390	0.353
KCF	0.204	0.098	0.222	0.139	0.125	0.191	0.135	0.237	0.221	0.245	0.283	0.232
DCF	0.205	0.099	0.224	0.148	0.126	0.190	0.133	0.237	0.223	0.246	0.286	0.234
DCF_CA	0.213	0.178	0.277	0.156	0.134	0.225	0.162	0.246	0.233	0.262	0.298	0.266
MCCT-H	0.343	0.242	0.415	0.243	0.172	0.337	0.245	0.341	0.357	0.391	0.438	0.368
CF2	0.339	0.273	0.415	0.228	0.223	0.363	0.211	0.340	0.340	0.377	0.420	0.370
SAMF_CA	0.269	0.185	0.354	0.250	0.151	0.245	0.182	0.317	0.272	0.328	0.355	0.314
SAMF	0.264	0.108	0.277	0.225	0.140	0.216	0.151	0.300	0.265	0.307	0.350	0.289
BACF	0.336	0.262	0.429	0.275	0.166	0.320	0.238	0.336	0.333	0.382	0.417	0.378
SRDCF	0.335	0.217	0.403	0.289	0.202	0.320	0.216	0.338	0.328	0.382	0.410	0.349
MUSTER	0.287	0.176	0.347	0.203	0.186	0.266	0.221	0.293	0.281	0.339	0.372	0.333
DSST	0.220	0.106	0.235	0.143	0.132	0.196	0.156	0.266	0.241	0.261	0.310	0.241
TLD	0.246	0.134	0.274	0.158	0.120	0.192	0.216	0.194	0.205	0.273	0.300	0.263
SPCF	0.401	0.290	0.480	0.318	0.228	0.406	0.295	0.360	0.399	0.439	0.479	0.439

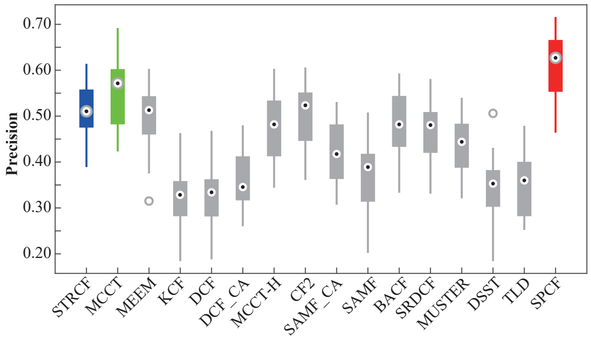


Fig. 8. The boxplot of 16 trackers on 12 attributes based on PP results. Red, green and blue fonts respectively represent 1st, 2nd, and 3rd performances among all trackers.

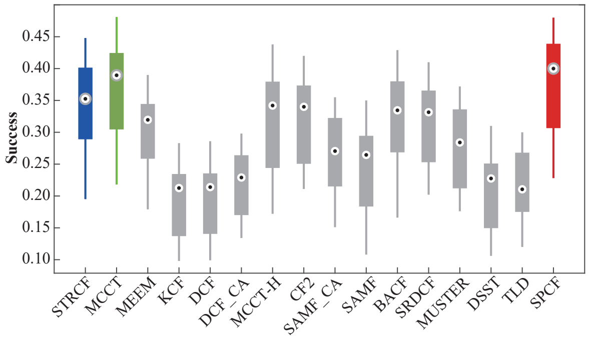


Fig. 9. The boxplot of 16 trackers on 12 attributes based on SP results. Red, green and blue fonts respectively represent 1st, 2nd, and 3rd performances among all trackers.

REFERENCES

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

[2] 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.

[3] M. Mueller, G. Sharma, N. Smith, and B. Ghanem, "Persistent Aerial Tracking system for UAVs," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1562–1569, 2016.

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

[5] 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," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5441–5446, 2014.

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

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on*

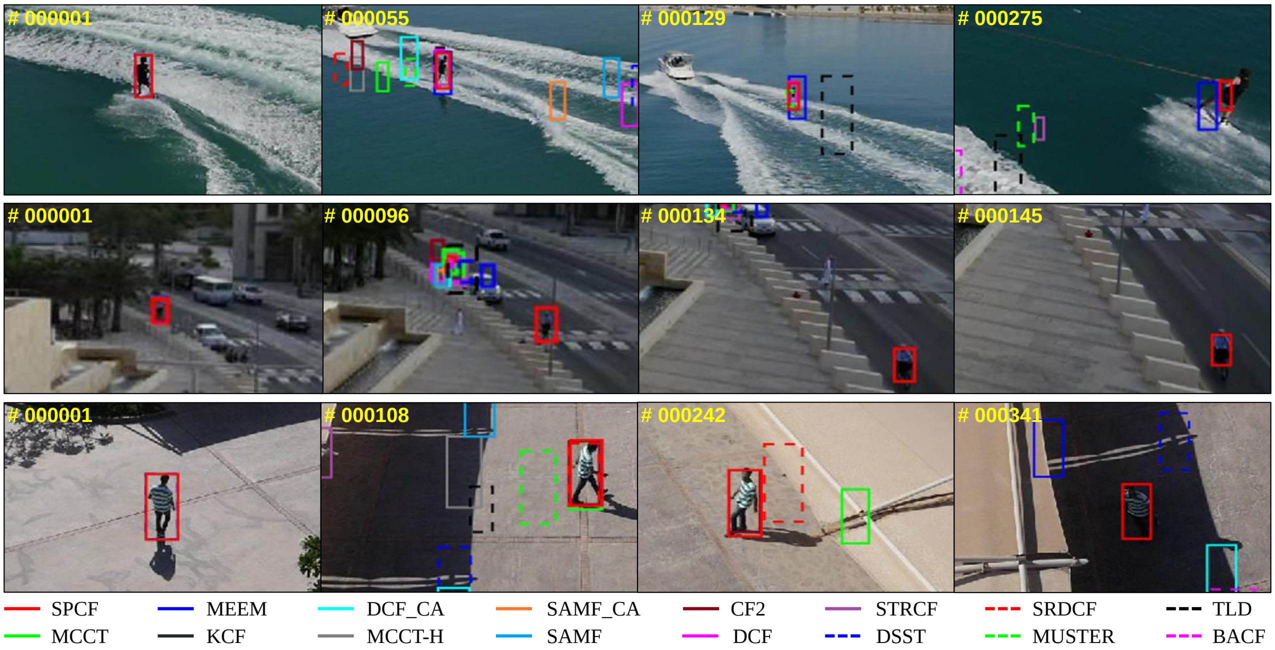


Fig. 10. Examples of the UAV tracking results. The first, second, and third columns show the image sequences from $wakeboard3\_1$, $bike3$ and $person12\_2$. Code and UAV tracking video are: `https://github.com/vision4robotics/SPCF-Tracker` and `https://youtu.be/7ccKeRi1hU8`.

Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583–596, 2015.

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

[9] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.

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

[11] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshop*, 2014, pp. 254–265.

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

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

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

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[16] J. V. de Weijer, C. Schmid, J. J. Verbeek, and D. Larlus, "Learning Color Names for Real-World Applications," *IEEE Transactions on Image Processing*, vol. 18, pp. 1512–1523, 2009.

[17] 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.

[18] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional Features for Correlation Filter Based Visual Tracking," *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop*, pp. 621–629, 2015.

[19] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-End Representation Learning for Correlation Filter Based Tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008, 2017.

[20] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "Dcfnet: Discriminant correlation filters network for visual tracking," *arXiv preprint arXiv:1704.04057*, 2017.

[21] "Transfer Learning Based Visual Tracking with Gaussian Processes Regression, author=Jin Gao and Haibin Ling and Weiming Hu and Junliang Xing, booktitle=Proceedings of the European Conference on Computer Vision (ECCV), year=2014."

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

[23] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2016, pp. 1430–1438.

[24] "Very Deep Convolutional Networks for Large-Scale Image Recognition, author=Karen Simonyan and Andrew Zisserman, journal=CoRR, year=2015, volume=abs/1409.1556."

[25] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 445–461.

[26] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 749–758.

[27] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1396–1404.

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

[29] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1135–1143.

[30] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4310–4318.