

# Augmented Memory for Correlation Filters in Real-Time UAV Tracking

Yiming Li<sup>1</sup>, Changhong Fu<sup>1,\*</sup>, Fangqiang Ding<sup>1</sup>, Ziyuan Huang<sup>2</sup>, and Jia Pan<sup>3</sup>

**Abstract**—The outstanding computational efficiency of discriminative correlation filter (DCF) fades away with various complicated improvements. Previous appearances are also gradually forgotten due to the exponential decay of historical views in traditional appearance updating scheme of DCF framework, reducing the model’s robustness. In this work, a novel tracker based on DCF framework is proposed to augment memory of previously appeared views while running at real-time speed. Several historical views and the current view are simultaneously introduced in training to allow the tracker to adapt to new appearances as well as memorize previous ones. A novel rapid compressed context learning is proposed to increase the discriminative ability of the filter efficiently. Substantial experiments on UAVDT and UAV123 datasets have validated that the proposed tracker performs competitively against other 26 top DCF and deep-based trackers with over 40 FPS on CPU.

## I. INTRODUCTION

Unmanned aerial vehicle (UAV) object tracking has many applications such as target tracing [1], robot localization [2], mid-air tracking [3] and aerial cinematography [4]. It aims to locate the object in the following frames given the initial location, in sometimes difficult situations such as fast motion, appearance variation (occlusion, illumination and viewpoint change, etc.), scale changes, and limited power capacity.

In UAV tracking tasks, the speed has been a key issue besides its performance. It was because of its ability to track objects at hundreds of frames per second (FPS) that discriminative correlation filter (DCF) is widely applied to perform UAV tracking in the first place. Unfortunately, the pioneering works [5]–[7], despite their incredible speed, have inferior tracking performances. Therefore, strategies like part-based methods [8], [9], spatial punishment [10]–[12] and robust appearance representation [13]–[15] are used to improve their precision and accuracy. However, speed of DCF is sacrificed in pursuit of better performances.

In order to adapt to appearance changes of tracked objects, an appearance model is maintained and updated at each frame for most DCF trackers. Due to its updating scheme, historical appearance decays exponentially with the number of subsequent frames. The appearance in latest 2 seconds in a 60 FPS video has a similar weight to all appearances before these 2 seconds in the model. This makes the trackers prone to forget objects’ early appearances and focus on more recent

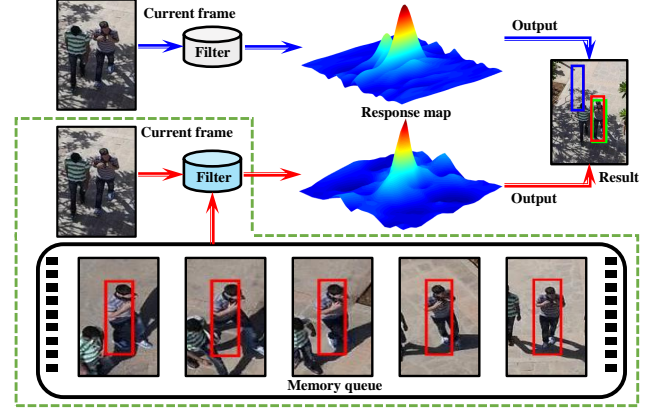


Fig. 1. Comparison between traditional DCF with appearance models and DCF with the proposed augmented memory. Multiple historical views are selected and stored to be used in training so that it contains more historical appearance information. Traditional DCF uses an gradually decaying appearance model and is prone to drift when drastic appearance variation happens. Red and blue boxes denote the tracking results of our tracker and others respectively. Ground truth is displayed as green box.

ones. Therefore, when the tracker has a false result, when the object is occluded, or when it is out of the view, it is very likely that the tracker learns appearances of the background using this scheme, which will further lead to lost of object in the following frames.

Additionally, traditional DCF framework has a low discriminative power because of the lack of negative training samples. Spatial regularization [10]–[12] and target cropping [16]–[19] were used to expand search region and extract background patches as negative samples. Introducing context and repressing response to it can also help discriminate objects from complex scenes [20]. These methods all propose effective ways to solve the problem, but not efficient ones.

This work proposes an augmented memory for correlation filters (AMCF) to perform efficient object tracking with strong discriminative power, which can easily be implemented on UAVs with only one CPU. Augmented memory is used to better memorize previous appearances of objects, with a novel application of image similarity criteria pHash [21] to carefully select what to memorize. Views in the memory and current view are simultaneously used in training the correlation filter so that it has suitable responses to both previous and current object appearances. Compressed context learning is proposed to rapidly learn the background so that discriminative power is efficiently raised. AMCF is evaluated extensively on the authoritative UAVDT and UAV123 datasets. The results show its competitive performance on CPU at over 40 FPS compared with top trackers.

<sup>1</sup>Yiming Li, Changhong Fu and Fangqiang Ding are with the School of Mechanical Engineering, Tongji University, 201804 Shanghai, China. changhongfu@tongji.edu.cn

<sup>2</sup>Ziyuan Huang is with the Advanced Robotics Centre, National University of Singapore, Singapore. ziyuan.huang@u.nus.edu

<sup>3</sup>Jia Pan is with the Computer Science Department, The University of Hong Kong, Hong Kong, China. panjia1983@gmail.com

## II. RELATED WORKS

### A. Real-time tracking for UAV using DCF

UAV object tracking has flourishing applications [22]–[26]. Different from generic object tracking operating in videos shot by stationary cameras, tracking for UAV requires trackers to perform robustly even with the drastic motion of on-board camera. Motion of drone camera combined with that of objects makes precise and robust tracking difficult. Since in most cases, the movement of the drone is going to take largely depends on its perception result, tracking for UAV also requires a high processing speed of the tracker.

Generative and discriminative methods have been applied to visual object tracking. One of the discriminative methods, DCF, has been widely adopted to perform the task. Although the original works [5]–[7] showed its exceeding performance in speed, they can hardly meet the requirement for accurate as well as robust tracking, and most current trackers have sacrificed their speed for better performances. A novel tracker that can balance the speed and performance is therefore called for.

### B. Model update scheme of DCF framework

One significant difference between DCF and deep-learning based tracking methods is that DCF can track objects online without any pre-training on appearances of the objects. This is achieved because DCF framework usually maintains an appearance model that is updated on a frame-to-frame basis [7], [10], [17]–[19], [27]. To do that, mostly adopted scheme is that a new model in the new frame is composed of around 99% of the previous model and around 1% of the new appearance. This 1% is treated as the learning rate. Some trackers use only new appearance as the new model [27]. Unfortunately, the updating scheme causes a model decay, which means the appearance in early frames only takes up a small weight in this model. Therefore, when occlusion, out-of-view, or lost track of object happens, trackers tend to learn appearances of the background. This is not robust enough.

### C. Negative samples and background noise in DCF

The efficiency of DCF stems from its transformation of correlation operations into frequency domain. To do that, object image is cyclically shifted and extracted as samples implicitly. For traditional DCF framework, the search area is limited to prevent the correlation filter to learn too much from the background. However, only positive sample is essentially exploited in this manner. Several measures are taken to expand search region and feed background patches to training as negative samples [10], [17]–[20]. Typically, [10] uses spatial punishment to suppress background learning, and [17] crops the target and background separately. [20] proposes to introduce context of the object and suppress the response to it. Despite their effectiveness, in order to be applied in UAV tracking, the efficiency of these methods is not sufficient.

### D. Tracking by deep learning

Deep learning is demonstrating its outstanding performance in various tasks. In DCF-based tracking, deep features are extracted by convolutional neural networks (CNN) in [13]–[15] to further improve performance by strengthening object appearance representation. In addition to DCF-based tracking methods, end-to-end learning [28], deep reinforcement learning [29], multi-domain network [30] and recurrent neural networks (RNNs) [31] directly use deep learning to perform tracking tasks. Despite their slightly superior performance, a high-end GPU is required for them to be trained and implemented. Even with that condition, most of them still run at low FPS. Therefore, deep-learning is not as suitable as DCF for aerial tracking tasks.

## III. REVIEW OF TRADITIONAL DCF

Learning traditional discriminative correlation filters [17] is to optimize the following function:

$$f(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2, \quad (1)$$

where  $\mathbf{X}$  is the sample matrix produced by circulating feature vector  $\mathbf{x} \in \mathbb{R}^{M \times N}$ ,  $\mathbf{w}$  is the trained filter and  $\lambda$  is for regularization. It uses a model update scheme as follows:

$$\mathbf{m}_t = (1 - \alpha)\mathbf{m}_{t-1} + \alpha\mathbf{x}_t, \quad (2)$$

where  $\mathbf{m}_t$  and  $\mathbf{x}_t$  denotes the model and object appearance feature in the  $t$ -th frame respectively,  $\alpha$  is the fixed learning rate. Early appearances of the object are decaying exponentially.

## IV. AUGMENTED MEMORY FOR DCF

In this section, learning augmented memory correlation filters for real-time UAV tracking is presented. The main structure of AMCF tracker is illustrated in Fig. 2, and the objective function is as follows:

$$f_t(\mathbf{w}) = \|\mathbf{X}_c\mathbf{w} - \mathbf{y}_c\|_2^2 + \lambda_2 \sum_{k=0}^K \|\mathbf{X}_k\mathbf{w} - \mathbf{y}_k\|_2^2 + \lambda_3 \|(\mathbf{H} \odot \mathbf{X}_b)\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_2^2, \quad (3)$$

where  $\mathbf{X}_c$ ,  $\mathbf{X}_k$  and  $\mathbf{X}_b$  presents the sample matrix generated by circulating feature maps  $\mathbf{x}_c$ ,  $\mathbf{x}_k$  and  $\mathbf{x}_b$ .  $\mathbf{x}_c$  represents the extracted patch in current frame,  $\mathbf{x}_0$  is the training patch sampled at the first frame,  $\mathbf{x}_k$  ( $k \in \{1, 2, \dots, K\}$ ,  $K \ll M$ ) is the  $k$ -th view from the memory queue introduced in IV-A, where  $M$  is the length of the sequence.  $\mathbf{y}_c$  and  $\mathbf{y}_k$  are distinct desired response of current frame and the ones in the memory (explained in IV-B).  $\mathbf{x}_b$  denotes compressed context patch,  $\mathbf{H}$  refers to a suppression matrix generated by circulating  $\mathbf{h} \in \mathbb{R}^{M \times N}$  and its function is to remove the object from the compressed context (explained in IV-C).  $\lambda_k$  ( $k = 1, 2, 3$ ) are adjustable parameters that determines the importance of corresponding patch.

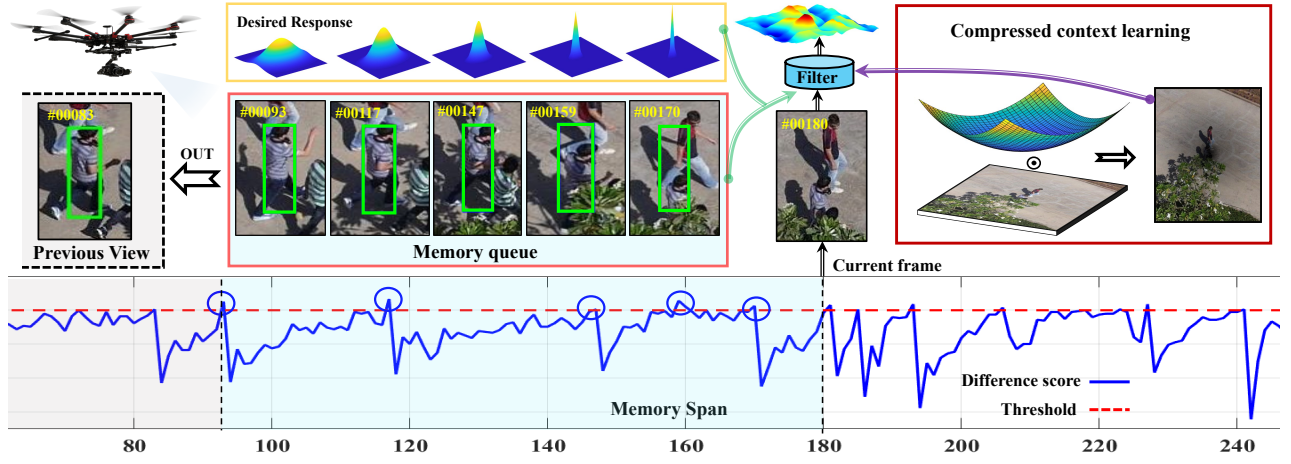


Fig. 2. **Overall structure of AMCF.** Several historical views are stored in the memory queue and assigned with different desired responses according to its distance to the current frame. The selection of view is based on a difference score calculated by perceptual hashing algorithm. Along with views in the memory queue, context is also used in training. Context of the current frame is first compressed and given a weight before it is fed into training process. Tracking code and video can be seen respectively: <https://github.com/vision4robotics/AMCF-tracker> and <https://youtu.be/CGH5o2Jl0hI>.

### A. Augmented memory

1) *Memory queue:* Basically, a first-in first-out queue is maintained with a total length of  $K$  to store  $K$  historical views, so that they can be exploited by the training of a correlation filter in each frame. Before the memory queue is full, i.e.,  $L < K$  (with  $L$  being the number of views currently stored), only  $L$  views are fed in the training process. Otherwise, all  $K$  historical views are used to train the correlation filter. For efficiency reasons, the value of  $K$  is significantly smaller than the total number of the sequence.

2) *View selection:* Since the number of views that can be stored is limited, it is important that different views contain different appearance details. Therefore, only when the appearance of object is significantly different than the last selected view is this appearance allowed into the memory queue. Perceptual hashing algorithm (PHA) [21] is adopted to determine the level of difference between two appearances. Specifically, the gray images are firstly transformed to frequency domain by discrete cosine transform (DCT). Only the low frequency region with high energy density is retained, denoted as  $\mathbf{B} \in \mathbb{R}^{S \times S}$ . Every element  $b_{ij}$  ( $i$  and  $j$  denote the index of the element) in  $\mathbf{B}$  is then compared with the average value of  $\mathbf{B}$  to generate respective element of  $p_{ij}$  in image's hashing matrix  $\mathbf{P} \in \mathbb{R}^{S \times S}$ :

$$\text{if } b_{ij} > \sum_{i=1}^S \sum_{j=1}^S b_{ij} / S^2 \text{ then } p_{ij} = 1; \text{ else } p_{ij} = 0. \quad (4)$$

Difference score of the last view and the current view is calculated using respective hashing matrices  $\mathbf{P}^l$  and  $\mathbf{P}^c$ :

$$\text{score} = \frac{\sum_{i=1}^S \sum_{j=1}^S (p_{ij}^c \oplus p_{ij}^l)}{S^2}, \quad (5)$$

where  $p_{ij}^c$  and  $p_{ij}^l$  respectively denote the element of hashing matrix  $\mathbf{P}^c$  and  $\mathbf{P}^l$ . And  $\oplus$  is the XOR operator. If the score is more than threshold  $\tau$ , two appearances are considered different and current one is selected into the memory queue.

### B. Different desired responses

Earlier selected views generally have a lower similarity to the current appearance than the later selected ones. Therefore, different desired responses are assigned to different views in the memory queue. By altering the maximum and the variance of Gaussian function, lower maximum and larger variance of desired responses are generated for early views:

$$\begin{aligned} \mathbf{y}_K(\sigma_K) &= \nu \mathbf{y}_c(\mu \sigma_c) \\ \mathbf{y}_k(\sigma_k) &= \nu \mathbf{y}_{k+1}(\mu \sigma_{k+1}) \quad (k = 1, 2, \dots, K-1) \end{aligned} \quad (6)$$

where  $\mathbf{y}_c(\mu \sigma_c)$  denotes the desired response of the current frame with the maximum of  $\max(\mathbf{y}_c(\mu \sigma_c))$  and the variance of  $\sigma_c$ . The subscript of  $\mathbf{y}$  represents the index value of response target in memory queue (lower values corresponds to earlier views). Parameters  $\nu < 1$  and  $\mu > 1$  make sure that maximum is decreasing and variance is increasing with the index decreasing. For the first frame of the sequence, the desired response is calculated as follows:

$$\mathbf{y}_0(\sigma_1) = \phi \mathbf{y}_c(\varphi \sigma_c), \quad (7)$$

where  $\phi < 1$  and  $\varphi > 1$  are used to adjust the maximum and variance of the Gaussian distribution of the desired response.

### C. Compressed context learning

In order to increase discriminative power of DCF, compressed context learning is proposed. Unlike traditional context learning, the enlarged search region is compressed to the size of the correlation filter. Then the pixel value where the object is located is lowered by applying a quadratic suppressing function, so as to remove the object from this patch. This compressed context is assigned zero response so that the response to the surrounding area of the object can be minimized and discriminative power can thus be enhanced.

### D. Learning and detection in AMCF

1) *Learning process:* The optimization result of  $f_t(\mathbf{w})$  for the  $d$ -th ( $d \in \{1, \dots, D\}$ ) channel is calculated as follows:

## V. EXPERIMENTS

In this section, the presented AMCF tracker is comprehensively evaluated on two difficult datasets, i.e., UAVDT [32] and UAV123 [33], with 173 image sequences covering over 140,000 frames captured by UAV in various challenging scenarios. It is noted that the videos from both datasets are recorded at 30 FPS. 11 real-time trackers (CPU based) are used to compare with AMCF, i.e., ECO\_HC [34], STRCF [27], MCCT\_H [35], STAPLE\_CA [20], BACF [17], DSST [36], fDSST [37], STAPLE [38], KCC [39], KCF [7], DCF [7]. Furthermore, 15 deep-based trackers are compared with AMCF to further demonstrate its performance, i.e., ASRCF [40], ECO [34], C-COT [11], MCCT [35], DeepSRTCF [27], ADNet [29], CFNet [28], MCPF [41], IBCCF [15], CF2 [14], CREST [42], HDT [43], FCNT [44], PTAV [45], TADT [46]. The evaluation criteria are strictly according to the protocol in two benchmarks [32], [33].

### A. Implementation details

All the experiments of all trackers compared as well as AMCF are conducted on a computer with an CPU of i7-8700K (3.7GHz), 48GB RAM and NVIDIA GTX 2080. All trackers are implemented in MATLAB R2018a platform, and their original codes without modification are used for comparison. Memory length  $K = 5$ , and the threshold for memory view selection is set to  $\tau = 0.5$ .

### B. Quantitative study

1) *Effectiveness study*: AMCF tracker is firstly compared with itself with different modules enabled. The effectiveness evaluation result can be seen in Table I. With each module (channel weight CW, augmented memory AM, and compressed context CC) added to the baseline, the performance is steadily being improved.

2) *Overall performance*: In comparison with other top real-time trackers, AMCF has shown a superiority in terms of precision and accuracy on both benchmarks. Fig. 3 shows separate performance evaluation of AMCF on two benchmarks. AMCF has achieved satisfactory performances on both benchmarks. Specifically, AMCF performs the best on UAVDT, with improvement of 0.6% and 1.2% on precision and AUC score respectively. On UAV123, AMCF has achieved the second best performance. Since the object size in UAV123 is generally much larger than that in UAVDT because of the flying height of UAVs, many trackers with

TABLE I

EFFECTIVENESS STUDY OF AMCF ON UAV123 AND UAVDT. MODULE NAME DISPLAYED IN ABBREVIATIONS ARE CW (CHANNEL WEIGHT), AM (AUGMENTED MEMORY) AND CC (COMPRESSED CONTEXT).

| Dateset        | UAV123 |      |      | UAVDT |      |      |
|----------------|--------|------|------|-------|------|------|
| Evaluation     | PREC.  | AUC  | FPS  | PREC. | AUC  | FPS  |
| AMCF           | 69.5   | 49.3 | 38.1 | 70.1  | 44.5 | 46.7 |
| Baseline+CW+CC | 68.5   | 48.3 | 42.6 | 67.3  | 44.0 | 53.1 |
| Baseline+CC+AM | 67.4   | 47.8 | 42.6 | 69.0  | 44.4 | 52.1 |
| Baseline+CW    | 66.8   | 46.8 | 50.4 | 67.9  | 44.0 | 65.5 |
| Baseline       | 65.5   | 46.8 | 59.1 | 66.4  | 42.9 | 73.5 |

$$\hat{\mathbf{w}}^d = \frac{\hat{\mathbf{M}}_c^d + \lambda_2 \sum_{k=0}^K \hat{\mathbf{M}}_k^d}{\sum_{d=1}^D (\hat{\mathbf{A}}_c^d + \lambda_2 \sum_{k=0}^K \hat{\mathbf{A}}_k^d + \lambda_3 \hat{\mathbf{E}}^d) + \lambda_1}, \quad (8)$$

where  $\hat{\mathbf{M}} = \hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}$ ,  $\hat{\mathbf{A}} = \hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}$  and  $\hat{\mathbf{E}} = \hat{\mathbf{m}}_b^* \odot \hat{\mathbf{m}}_b$ .  $\mathbf{m}_b = \mathbf{h} \odot \mathbf{x}_b$ .  $\mathbf{x}^*$  denotes the complex-conjugate of  $\mathbf{x}$  and the operator  $\odot$  stands for the element-wise product. Hat mark is the discrete Fourier transform (DFT) value.

A learning rate  $\gamma$  is used to update the numerator  $\hat{\mathbf{N}}_t^d$  and the denominator  $\hat{\mathbf{D}}_t^d$  of the filter  $\hat{\mathbf{w}}_t^d$  in the  $t$ -th frame:

$$\begin{aligned} \hat{\mathbf{w}}_t^d &= \frac{\hat{\mathbf{N}}_t^d}{\sum_{d=1}^D \hat{\mathbf{D}}_t^d + \lambda_1} \\ \hat{\mathbf{N}}_t^d &= (1 - \gamma) \hat{\mathbf{N}}_{t-1}^d + \gamma (\hat{\mathbf{M}}_{tc}^d + \lambda_2 \sum_{k=0}^K \hat{\mathbf{M}}_{tk}^d) \\ \hat{\mathbf{D}}_t^d &= (1 - \gamma) \hat{\mathbf{D}}_{t-1}^d + \gamma (\hat{\mathbf{A}}_{tc}^d + \lambda_2 \sum_{k=0}^K \hat{\mathbf{A}}_{tk}^d + \lambda_3 \hat{\mathbf{E}}_t^d) \end{aligned} \quad (9)$$

In order to make sure reliable channels contribute more to the final result, a channel weight  $\mathbf{C} = \{c^d\} (d \in \{1, \dots, D\})$  is assigned to each channel and updated as follows:

$$c_t^d = (1 - \eta) c_{t-1}^d + \eta \frac{\max(\hat{\mathbf{w}}_t^{d*} \odot \hat{\mathbf{x}}_{tc}^d)}{\sum_{d=1}^D \max(\hat{\mathbf{w}}_t^{d*} \odot \hat{\mathbf{x}}_{tc}^d)}. \quad (10)$$

2) *Detection in AMCF*: In detection phase, the following formula is used to generate the final response map  $\mathbf{R}_t$  and update the position by searching the maximum value:

$$\mathbf{R}_t = \mathcal{F}^{-1} \left( \sum_{d=1}^D c_t^d \hat{\mathbf{w}}_{t-1}^{d*} \odot \hat{\mathbf{z}}_t^d \right), \quad (11)$$

where  $\hat{\mathbf{w}}^{d*}$  and  $\hat{\mathbf{z}}_t^d$  are respectively learned filter and current feature of search region in frequency domain, and  $\mathcal{F}^{-1}$  denotes the inverse discrete Fourier transformation (IDFT).

---

#### Algorithm 1: AMCF tracker

---

**Input:** Groundtruth in the first frame  
Subsequent frames  
**Output:** Predicted position of target in  $t > 1$  frame

```

1 if  $t = 1$  then
2   Extract  $\mathbf{x}_f$  and  $\mathbf{x}_b$  centered at the groundtruth
3   Use Eq. (8) to initialize the filters  $\mathbf{w}_1$ 
4   Initialize channel weight model  $\{c^d\} = 1/D$ 
5 else
6   Extract  $\mathbf{z}_t$  centered at location on frame  $t - 1$ 
7   Use Eq. (10) to generate the response map
8   Find the peak position of map and output
9   Extract  $\mathbf{x}_t$  and  $\mathbf{x}_b$  centered at location on frame  $t$ 
10  Calculate the score between  $\mathbf{P}_L$  and  $\mathbf{P}_t$ 
11  if  $\text{score} > \tau$  then
12    Update FIFO memory queue
13  end
14  Use Eq. (8) to update the filters  $\mathbf{w}_t$ 
15  Use Eq. (9) to update channel weight  $\mathbf{C}_t$ 
16 end
```

---



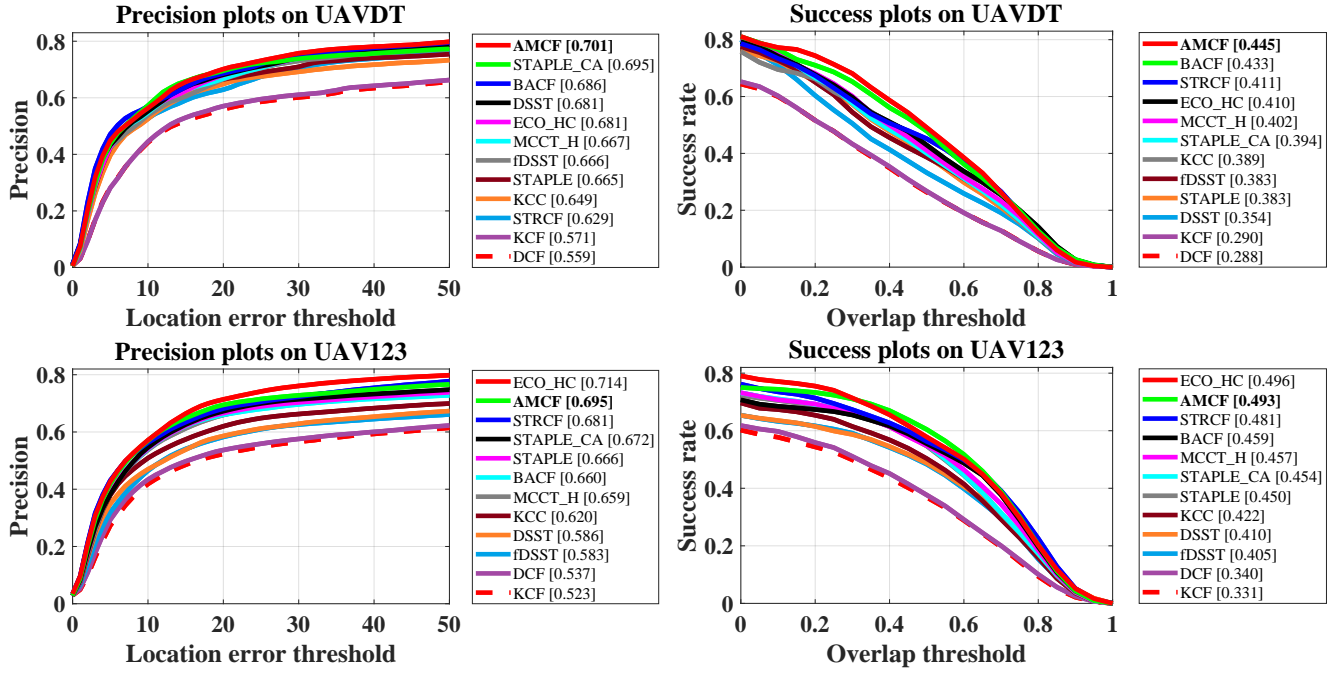


Fig. 3. **Overall performance evaluation.** Precision and success plots of our tracker and other ten top real-time trackers on UAVDT and UAV123 datasets.

TABLE II

AVERAGE FRAME PER SECOND (FPS) AND AVERAGE PRECISION AS WELL AS AUC OF TOP REAL-TIME TRACKERS ON 173 IMAGE SEQUENCES. **RED**, **GREEN** AND **BLUE** FONTS INDICATE THE FIRST, SECOND AND THIRD PLACE, RESPECTIVELY. ALL RESULTS ARE OBTAINED SOLELY ON CPU.

| Tracker   | AMCF        | ECO_HC      | MCCT_H | STAPLE_CA   | KCC  | BACF        | STRCF       | STAPLE | DSST  | fDSST        | KCF          | DCF           |
|-----------|-------------|-------------|--------|-------------|------|-------------|-------------|--------|-------|--------------|--------------|---------------|
|           |             | [34]        | [35]   | [20]        | [39] | [17]        | [27]        | [38]   | [36]  | [34]         | [7]          | [7]           |
| FPS       | 42.4        | 77.4        | 59.9   | 60.2        | 48.9 | 58.8        | 29.8        | 84.5   | 124.1 | <b>186.1</b> | <b>795.7</b> | <b>1120.7</b> |
| Precision | <b>69.8</b> | <b>69.8</b> | 66.1   | <b>67.7</b> | 62.6 | 66.6        | <b>67.0</b> | 66.6   | 60.7  | 60.1         | 53.3         | 54.2          |
| AUC       | <b>46.9</b> | <b>45.3</b> | 43.0   | 42.4        | 40.6 | <b>44.6</b> | <b>44.6</b> | 41.7   | 38.2  | 39.4         | 31.1         | 34.4          |

good performance on one benchmark can rank low on the other. One typical example is ECO\_HC. It ranks first on UAV123 but only come out at fifth place on UAVDT. AMCF, on the other hand, has a better generalization ability compared to most trackers. Overall evaluation for both benchmarks combined can be seen in Table II. AMCF has a slightly better overall performance than ECO\_HC, ranking the first place. But ECO\_HC has a relatively large variance. Therefore, it can be concluded that in terms of overall performance, AMCF performs favorably against other top real-time trackers. In terms of speed, AMCF, solely running on CPU, can also meet the requirement of real-time tracking (>30 FPS on a image sequence captured at 30 FPS).

3) *Attribute-based performance:* Attribute-based evaluation results on both benchmarks are shown in Fig. 4. It can be seen that the reason behind our satisfactory generalization ability originates from the combination of two of our core modules, i.e., augmented memory and compressed context learning. On UAVDT, when object is small, feature of objects decreases and positive samples are thus not enough for robust tracking. Compressed context learning simultaneously expands search region and brings more negative samples. Therefore, when there is camera motion and background clutter, AMCF performs satisfactorily. On UAV123, object

is significantly closer, so viewpoint change and aspect ratio change can result in more drastic appearance changes. Augmented memory provides more appearance information on previous objects so that a desired response can be obtained when current view has some resemblance to previous views. Therefore, thanks to both modules, ARCF can handle both near objects with large viewpoint changes and distant objects with a small size on the image.

### C. Comparison with deep-based trackers

On UAVDT, extra 15 deep-based trackers (trackers using deep learning methods or DCF-based trackers using deep features) are compared with AMCF. Precision and success plots can be seen in Fig. 5. Surprisingly, AMCF still succeeds to exceed most of deep-based trackers in terms of precision and AUC scores. More specifically, AMCF achieved the best performance in precision, with 0.1% slightly better than the second place ASRCF, while in AUC evaluation, AMCF achieved the second, falling behind ECO by only 0.9%. In terms of tracking speed, AMCF is the fastest among the evaluated deep-based trackers. To sum up, in tracking distant objects, AMCF demonstrates superior tracking performance against both real-time trackers and deep-based trackers.

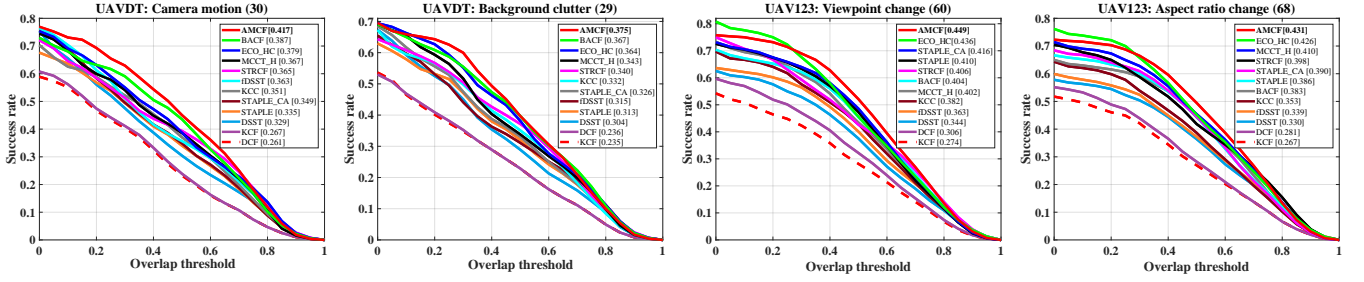


Fig. 4. **Attribute-based evaluation.** Success plots of four attributes. The first two attributes are from UAVDT and the rest of them are from UAV123.

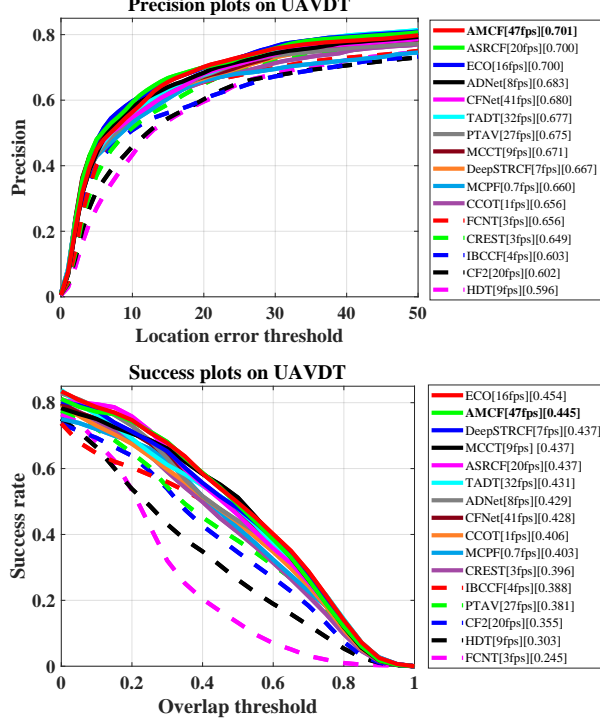


Fig. 5. **Deep-based tracker comparison.** Top deep-based trackers are compared with AMCF on UAVDT. Tracking speeds other than AMCF are obtained on GPU.



Fig. 6. **Qualitative evaluation.** Top real-time trackers are compared with AMCF on *S1001* and *S1201* on UAVDT, as well as *wakeboard5*, *car16\_1*, *car7*, and *car1\_s* on UAV123.

#### D. Qualitative study

Figure 6 intuitively demonstrates the aforementioned results. The first two sequences show the ability to track distant objects and adapt to view changes on UAVDT respectively. The third and fourth sequences show those abilities of AMCF on UAV123. Capability of resisting occlusion is demonstrated in the fifth sequence. The last one shows two modules can work smoothly together.

### VI. CONCLUSIONS

In this work, augmented memory for correlation filters is proposed. Essentially, augmented memory maintains a FIFO queue to store distinct previous views. Each stored view in the memory will be assigned a desired response. Along with the feature of the object in the current frame, all views are simultaneously used to train a correlation filter that can adapt to new appearances and has response to previous views at the

same time. Compressed context learning provides more negative samples and suppresses responses to surrounding areas of the tracked object. Extensive experiment results proved that AMCF has competitive performance and tracking speed compared to top real-time trackers. AMCF also demonstrates an outstanding generalization power that can track both near objects with large view change and distant objects with small size in the image. Future work can include introducing a confidence check to prevent false tracking results to be selected as a view. This method can also be applied to more powerful baselines in replacement of model update.

### ACKNOWLEDGMENT

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

## REFERENCES

- [1] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 446–453.
- [2] B. Coltin, J. Fusco, Z. Moratto, O. Alexandrov, and R. Nakamura, "Localization from visual landmarks on a free-flying robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4377–4382.
- [3] C. Fu, A. Carrio, M. A. Olivares-Méndez, R. Suarez-Fernandez, and P. C. Cervera, "Robust Real-time Vision-based Aircraft Tracking From Unmanned Aerial Vehicles," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5441–5446.
- [4] B. Rogerio, H. Cherie, W. Wenshan, C. Sanjiban, and S. Sebastian, "Can a Robot Become a Movie Director? Learning Artistic Principles for Aerial Cinematography," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
- [5] 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.
- [6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2012, pp. 702–715.
- [7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, March 2015.
- [8] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912.
- [9] C. Fu, Y. Zhang, R. Duan, and Z. Xie, "Robust Scalable Part-Based Visual Tracking for UAV with Background-Aware Correlation Filter," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018, pp. 2245–2252.
- [10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning Spatially Regularized Correlation Filters for Visual Tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4310–4318.
- [11] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking," in *European Conference on Computer Vision*, 2016, pp. 472–488.
- [12] A. Lukešić, T. Vojić, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4847–4856.
- [13] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional Features for Correlation Filter Based Visual Tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 621–629.
- [14] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical Convolutional Features for Visual Tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.
- [15] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M. Yang, "Integrating Boundary and Center Correlation Filters for Visual Tracking with Aspect Ratio Variation," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 2001–2009.
- [16] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Correlation Filters With Limited Boundaries," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [17] 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.
- [18] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *ICCV*, 2019.
- [19] C. Fu, Z. Huang, Y. Li, R. Duan, and P. Lu, "Boundary Effect-Aware Visual Tracking for UAV with Online Enhanced Background Learning and Multi-Frame Consensus Verification," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [20] M. Mueller, N. Smith, and B. Ghanem, "Context-Aware Correlation Filter Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1387–1395.
- [21] S. S. Kozat, R. Venkatesan, and M. K. Mihcak, "Robust perceptual image hashing via matrix invariants," in *2004 International Conference on Image Processing, 2004. ICIP '04.*, vol. 5, Oct 2004, pp. 3443–3446 Vol. 5.
- [22] C. Fu, A. Carrio, M. A. Olivares-Mendez, and P. Campoy, "Online learning-based robust visual tracking for autonomous landing of Unmanned Aerial Vehicles," in *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 649–655.
- [23] 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.
- [24] H. Cheng, L. Lin, Z. Zheng, Y. Guan, and Z. Liu, "An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1732–1738.
- [25] F. S. Leira, T. A. Johansen, and T. I. Fossen, "A UAV ice tracking framework for autonomous sea ice management," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 581–590.
- [26] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [27] F. Li, C. Tian, W. Zuo, L. Zhang, and M. Yang, "Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913.
- [28] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2805–2813.
- [29] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1349–1358.
- [30] H. Nam and B. Han, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [31] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1449–1458.
- [32] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: object detection and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, Conference Proceedings, pp. 370–386.
- [33] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2016, pp. 445–461.
- [34] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6931–6939.
- [35] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue Correlation Filters for Robust Visual Tracking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4844–4853.
- [36] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.
- [37] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative Scale Space Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.
- [38] 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.
- [39] C. Wang, L. Zhang, L. Xie, and J. Yuan, "Kernel cross-correlator," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [40] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4670–4679.
- [41] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4335–4343.
- [42] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, Conference Proceedings, pp. 2555–2564.
- [43] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, Conference Proceedings, pp. 4303–4311.
- [44] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3119–3127.
- [45] H. Fan and H. Ling, "Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5486–5494.
- [46] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.