Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Robust multi-kernelized correlators for UAV tracking with adaptive context analysis and dynamic weighted filters

Changhong Fu · Yujie He · Fuling Lin · Weijiang Xiong

Received: date / Accepted: date

Abstract In recent years, the correlation filter (CF)-based method has significantly advanced in the tracking for unmanned aerial vehicles (UAV). As the core component of most trackers, CF is a discriminative classifier to distinguish the object from the surrounding environment. However, the poor representation of the object and lack of contextual information have restricted the tracker to gain better performance. In this work, a robust framework with multi-kernelized correlators is proposed to improve robustness and accuracy simultaneously. Both convolutional features extracted from the neural network and hand-crafted features are employed to enhance expressions for object appearances. Then, the adaptive context analysis strategy helps filters to effectively learn the surrounding information by introducing context patches with the GMSD index. In the training stage, multiple dynamic filters with time-attenuated factors are introduced to avoid tracking failure caused by dramatic appearance changes. The response maps corresponding to different features are finally fused before the novel resolution enhancement operation to increase distinguishing capability. As a result, the optimization problem is reformulated, and a closed-form solution for the proposed framework can be obtained in the kernel space. Extensive experiments on 100 challenging UAV tracking sequences demonstrate the proposed tracker outperforms other 23 state-of-the-art trackers and can effectively handle unexpected appearance variations under the complex and constantly changing working conditions.

Changhong Fu (✉), Yujie He, Fuling Lin, Weijiang Xiong
School of Mechanical Engineering, Tongji University, 201804 Shanghai, China
Tel.: +86-137-6166-0140
E-mail: changhongfu@tongji.edu.cn

Keywords Visual tracking · Unmanned aerial vehicle (UAV) · Multi-kernelized correlators · Adaptive context analysis · Dynamic weighted filters

1 Introduction

The breakthrough in intelligent vision-based techniques for unmanned aerial vehicles (UAV) has been paid more attention recently and sparked a wide range of object tracking methods to tackle specific issues. In practice, the visual tracking for UAV can be applied in many fields, e.g., wildlife monitoring [28] and autonomous landing [23]. Considerable progress in correlation filter (CF)-based tracking methods has been made in the last several years. However, object tracking for UAV remains thorny due to dramatic object appearance changes. These issues are raised by several challenging factors [36, 26], including object deformation, illumination variation, partial or full occlusion, similar objects, and cluttered background. Additionally, the operation in mid-air brings unique difficulties such as mechanical vibrations, which severely degrade the performance.

The current methods are still limited in some aspects. One major weakness is that the search region of most trackers only contains a small neighboring area around the object to ensure low computational cost. According to the circulant assumptions, boundary effects inevitably arise by introducing the unreal samples in the training process, which decreases the robustness of the tracker. Furthermore, most proposed tracking approaches discard the continuously updated filters. They merely use the information from the latest frame to update the model, resulting in limited knowledge of the historical information and easily getting drifted when fast motion or occlusion happens.

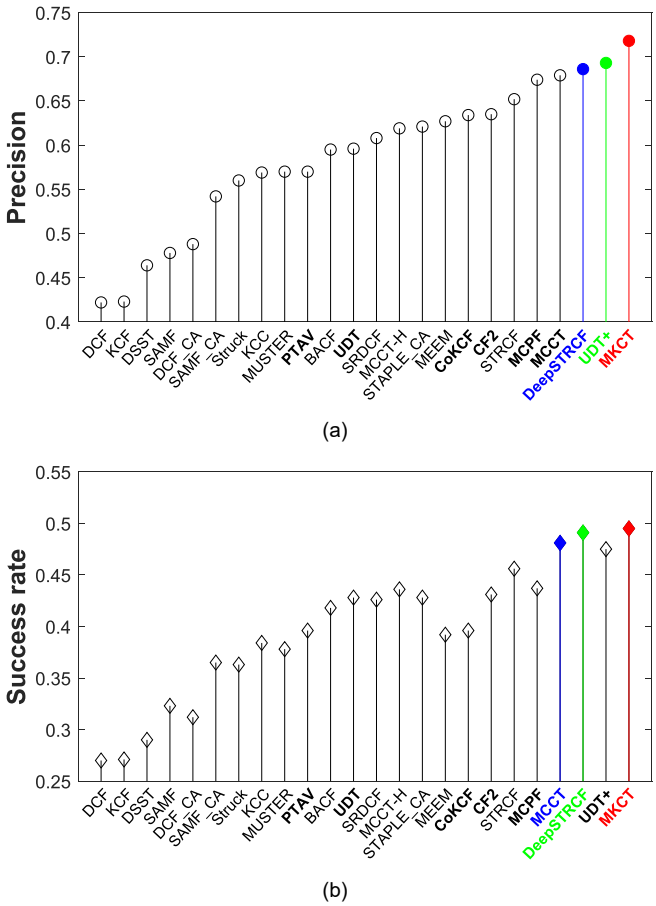

Fig. 1 Evaluation results of the proposed and state-of-the-art trackers. (a) and (b) display the precision and success rate on the challenging UAV image dataset, respectively. Red, green, and blue denote the first, second, and third best performance among all trackers. In addition, the trackers with **bold** fonts are the trackers based on deep learning

In the literature, some image measurements, e.g., luminance, contrast, and local structure, are shifting over time, especially on the challenging UAV sequences. According to different analysis theories, two types of full-reference image quality metrics are applied in computer vision applications [29, 35, 25]. On the one hand, the mathematical-based image quality index, such as minimum mean square error, can be applied in similarity assessment while these objective methods are vulnerable to appearance variation [42]. On the other hand, studies conducted in [16, 37] demonstrate that human vision system based metrics achieve much better performance than the aforementioned metrics. Inspired by the applications of image quality assessment in the generative tracking method [10], an image quality index is introduced in this work specifically for UAV visual tracking. The image quality measurement-based scheme is used to measure the similarity of different context patches referring to the object, thus improving the generalization to imaging scenarios adaptively.

In this work, a novel tracking approach, i.e., MKCT tracker, is proposed for UAV tracking to achieve advanced performance. It includes robust multi-kernelized correlators with adaptive context analysis and dynamic weighted filters. The similarity-based context learning scheme is devised to fully utilize contextual information to reinforce the sensitivity of the kernelized correlators. By introducing multiple dynamic filters, the model can be trained with historical information to avoid drift. Moreover, a resolution enhancement operation is used to develop anti-interference ability by weakening noise and sharpening the principal peak in the final response map. The contributions of the proposed method are listed as follows:

- A new tracking framework with multi-kernelized correlators is introduced. In the training stage, different features are employed with kernel methods and mapped into high-dimensional space, which improves the encoding ability of the model.
- A novel adaptive context analysis scheme is developed. It is the first time for the gradient magnitude similarity deviation (GMSD) [37] to be employed in object visual tracking as an image quality index. Combined with captured surrounding patches, the GMSD-based scheme can be used as a weighting measurement referring to the object. Thus, contextual information is thoroughly exploited to construct adaptive distractors and achieve better robustness.
- A dynamic learning strategy is developed to employ multiple filters. The influence of filters to different degrees are taken into account by assigning time-attenuated factors, which helps avoid over-fitting in the training stage and achieve better performance against dramatic appearance variations.
- A resolution enhancement operation for response maps of different features is proposed to improve the sensitivity in the detection stage. The output of the correlators is more accurate and robust with suppressed noise information and sharped main peaks during operation.

The proposed tracking approach is adequately evaluated and performs favorably against 23 other state-of-the-art trackers, as shown in Fig. 1. To the best of the knowledge, the proposed tracking method, i.e., MKCT tracker, has not been designed and employed for UAV tracking applications in the literature.

The remainder of this work is structured as follows: Section 2 covers the related works. Section 3 introduces the details of the proposed MKCT tracker. Sec-

tion 4 shows qualitative and quantitative experiment results and comparison with other state-of-the-art trackers. Section 5 presents conclusions and outlook for the future work.

2 Related works

2.1 Tracking with correlation filters

The CF-based trackers have been widely employed for object tracking owing to high computational efficiency. Many trackers, including the minimum output sum of squared error [2], i.e., MOSSE tracker, kernelized correlation filters [13, 14], discriminative scale space tracking [4], co-trained kernelized correlation filters [41] and many other trackers [22, 5, 24, 33, 32] have applied CF framework. Nevertheless, the CF-based methods have a drawback in lacking enough knowledge of the surrounding environment, which easily leads to drift on the challenging UAV sequences. Efforts have been made to improve the utilization of contextual information. For example, background-aware correlation filter (BACF) [18] exploits background patches by using the cropping operator to add real samples instead of synthetic ones. Context-aware correlation filter (CACF) [27] applied context patches into the training stage. With fixed weights to diverse patches of information, these trackers lack the generalization ability in the cluttered environment. Besides, the poor encoding ability also limits the distinguishing power of the tracker. More recently, kernel cross-correlator (KCC) is presented in [32] to learn kernelized filters to promote the representation of the sample. However, the lack of context information and multiple kernelized correlators limit the discriminative ability of the tracker.

2.2 Tracking with multiple features

The performance of the tracker is highly dependent on the expression of the tracking object. Two types of features, i.e., hand-crafted features and convolutional features, are widely used for object tracking. Hand-crafted features including histograms of gradients (HOG) [9] and color names (CN) [31] have been employed separately [5, 4, 14] or in combination with different hand-crafted features [1, 22] to enhance appearance representation. Despite their speeds, the trackers employing hand-crafted features focus on the shallow appearance model solely so that they are vulnerable in complex UAV tracking environment. Compared to hand-crafted features, using deep features from convolutional neural networks can remarkably increase the robustness of the trackers, which encouraged increasing applications in visual tracking methods recently. Danelljan M et al. [6] utilized different spatial information of the convolutional features for training filters. Ma C et al. [24] proposed a tracker to represent the tracking object with multi-level features from convolutional neural networks (CNN) to obtain both spatial and semantic information. In [7], a learning discriminative convolution operators based on in the continuous spatial domain is proposed. However, these trackers mainly rely on the appearance model of the object and ignore surrounding information. In this work, both ensembled hand-crafted and convolutional features are utilized to exploit both appearance and semantic information for representing the object and contextual patches. With multiple features, the proposed tracking approach can obtain optimum performance in the complicated UAV scenarios.

2.3 Tracking with similarity metric

Under the complicated situations, it is difficult to cope with serious issues caused by cluttered information. By utilizing the context patches from surrounding information, the object can be stressed to distinguish the object from the background. Therefore, the similarity metrics between the two samples can be employed to adopt as weighted factors to exploit contextual information fully. In recent years, different similarity including Euclidean distance [40], Mahalanobis distance [38], cross-bin metric [19] are applied in tracking approaches. Some weighting metrics [40, 38] are fixed without the online update, while other methods [19, 10] can be adaptive to the object appearance changes. However, these generative methods may result in tracking failure when distractors from the background bring interference information or share a similar appearance model with the object. Different from existing methods [27, 18], the GMSD index can effectively assess the similarity between each context patch and the object by capturing the local spatial characteristic. Above all, the adaptive GMSD-based context analysis scheme can be incorporated with a multi-kernelized correlators framework to achieve superior performance on the occurrence of the background clutter.

3 Proposed tracking method

3.1 Overview

In this section, the proposed robust multi-kernelized correlators for UAV tracking with adaptive context analysis and dynamic weighted filters, i.e., MKCT tracker,

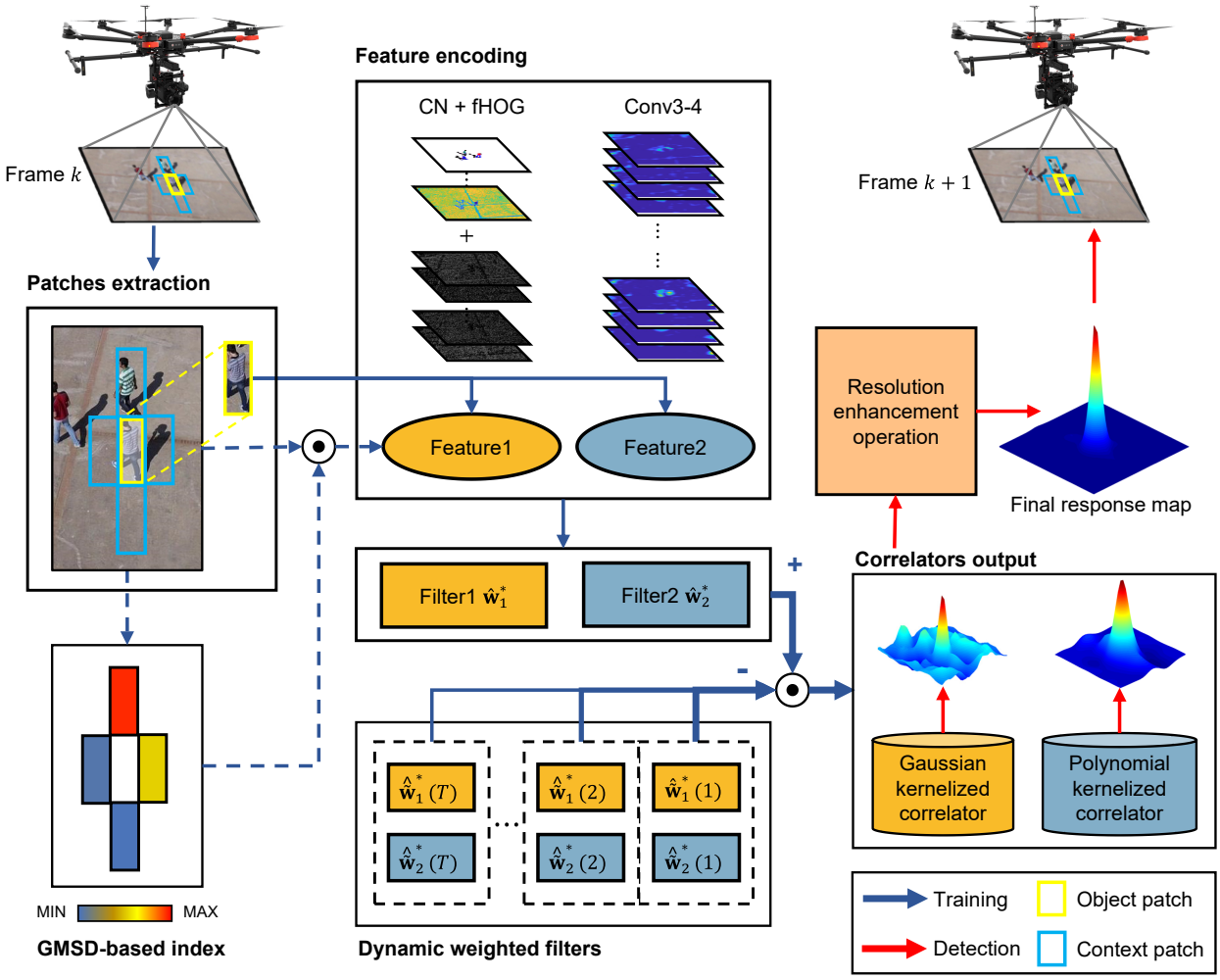

Fig. 2 The main workflow of the proposed MKCT tracker. The object patch is extracted every frame while the context patches near the object are extracted every Δ_c frames ($\Delta_c = 10$ in this work) from the search areas, which are obtained by the predicted position in the previous frame. The extracted patches are encoded by ensembled hand-crafted features as well as convolutional features before adaptive GMSD-based context analysis. Further, the model is trained with dynamic weighted filters jointly to construct different kernelized correlators. As a result, the response map corresponding to each correlator is processed by the resolution enhancement operation to sharpen the peak. Finally, the tracking result can be achieved by searching for the maximum value in the enhanced response map. Additionally, $\hat{\tilde{\mathbf{w}}}_n^*(1)$, $\hat{\tilde{\mathbf{w}}}_n^*(2)$ and $\hat{\tilde{\mathbf{w}}}_n^*(T)$ denote the 1st, 2nd, and T-th latest selected filters in filter pool

are introduced. The main steps of the proposed tracking method are shown in Fig. 2. From the start, the search window of the object is extracted from the previously estimated location and updated every frame. Additionally, the neighboring four patches with adaptive GMSD-based importance are selected and added to filters training at the interval. Accordingly, ensembled hand-crafted features (fHOG [9] and CN [31]) and deep features extracted from VGG-Net [30], are utilized for encoding the object and its surrounding information. Moreover, they are incorporated into the framework to generate kernelized correlators in different kernel space. Besides, the multiple dynamic weighted filters are joined in the training stage as external restraints to avoid drift in the complex scenarios. Therefore, the optimization problem is reformulated, and a closed-form solution can be achieved accurately and robustly concurrently. After obtaining response maps, the outputs of the two kernelized correlators are fused and processed by the proposed resolution enhancement operator to increase distinguishing power.

3.2 Multi-kernelized correlators framework

The novel proposed multi-kernelized correlators framework is formulated and based on the kernel method. For simplicity, the vectorized image can be denoted as column vectors $\mathbf{x}$, $\mathbf{z} \in \mathbb{R}^M$. With a non-linear kernel

function $\varphi(\cdot) : \mathbb{R}^M \to \mathbb{R}^H, H \gg M$, the inner product between $\mathbf{x}$ and $\mathbf{z}_i$ can be mapped into high-dimensional space. Then, kernelized correlator can be denoted by:

$$\kappa(\mathbf{x}, \mathbf{z}_i) = \varphi(\mathbf{x})^\mathsf{T} \varphi(\mathbf{z}_i) \ , \tag{1}$$

where the superscript $(\cdot)^\mathsf{T}$ denotes the transpose of a vector, and further $\kappa(\mathbf{x}, \mathbf{z}_i) \in \mathbb{R}$ avoids calculation redundancy for high dimensional features. Besides, the sample-based vector $\mathbf{z}_i \in \mathbb{R}^M$ is generated from the test sample $\mathbf{z}$ by the transform function $\mathcal{T}(\cdot)$, i.e., $\mathbf{z}_i \in \mathcal{T}(\mathbf{z})$. As a result, the sample-based vectors set can construct the kernel vector $\mathbf{k}^{\mathbf{xz}} = [k_1^{\mathbf{xz}}, \cdots, k_n^{\mathbf{xz}}]^\mathsf{T}$, where $k_i^{\mathbf{xz}}$ is used to represent $\kappa(\mathbf{x}, \mathbf{z}_i)$. Hence, the kernelized cross-correlation output can be denoted by:

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

where $\odot$ and superscript $(\cdot)^*$ denotes the element-wise product and complex conjugate operation, respectively. The superscript $\hat{\cdot}$ indicates the discrete Fourier transform of a vectorized image, i.e.:

$$\hat{\mathbf{x}} = \mathcal{F}\mathbf{x} \ . \tag{3}$$

Finally, the pattern of the training sample will be encoded in $\mathbf{w}$, and training sample $\mathbf{x}$ can be predicted as a result of the correlation output $\hat{C}(\mathbf{x}, \mathbf{z})$.

Suppose that N features are utilized to the represent tracking object. Accordingly, the multi-kernelized correlators framework, as well as dynamic weighted filters in the Fourier domain, can be formed by minimizing the regression target:

$$\begin{aligned}\hat{\mathcal{E}}(\hat{\mathbf{w}}^*) = \sum_{n=1}^{N} \Big(& \big\|\hat{C}_n(\mathbf{x}_{n0}, \mathbf{x}_{n0}) - \hat{\mathbf{y}}_n\big\|_2^2 + \lambda_1 \big\|\hat{\mathbf{w}}_n^*\big\|_2^2 \\ & + \lambda_2 \sum_{s=1}^{S} \Big\| f_{ns} \hat{C}_n(\mathbf{x}_{ns}, \mathbf{x}_{ns}) \Big\|_2^2 \\ & + \sum_{t=1}^{T} \Big\| \gamma_t [\hat{\mathbf{w}}_n^* - \hat{\tilde{\mathbf{w}}}_n^*(t)] \Big\|_2^2 \Big)\end{aligned} \ , \tag{4}$$

where $\mathbf{x}_{n0}$ and $\mathbf{x}_{ns}$ are the vectorized image patch of the object and context patches, respectively in the n-th feature. Then, $\hat{\mathcal{E}}$ is an error measured by the correlation output of $\mathbf{x}_{n0} \in \mathbb{R}^M$ and desired output $\mathbf{y}_n \in \mathbb{R}^M$. $\hat{\mathbf{w}}_n^* \in \mathbb{R}^M$ denotes the correlation filter corresponding to n-th feature in the Fourier domain. λ_1 and λ_2 are regularization factors to control the filters and correlation output of context patches, respectively. S is the number of context patches which are extracted from the top, bottom, left, and right directions close to the object. These patches are considered as hard negative samples, so their desired correlation output to each sample is zero. Then, a GMSD-based importance factor f_{ns} is proposed to evaluate the different importance of context patches at an interval Δ_c. Consequently, T is the number of selected weighted filters $[\hat{\tilde{\mathbf{w}}}_n^*(1), \cdots, \hat{\tilde{\mathbf{w}}}_n^*(T)]$ and γ_t is the penalty factor with time-attenuated property to construct the dynamic restraints and update at an interval Δ_f to prevent drift when subjected to appearance changes dramatically.

Remark 1: Different correlators are trained and updated in the kernel space of different dimensions, i.e., the samples represented by the ensembled hand-crafted features are trained with the Gaussian kernel function, and the samples represented by the deep features are trained with the polynomial kernel. Besides, λ_2 is a factor used to construct adaptive context constraints.

Because of the mutual independence of the equations corresponding to different features, the original objective function $\hat{\mathcal{E}}(\hat{\mathbf{w}}^*)$ in Eq. (4) can be reformulated subproblem $\hat{\mathcal{E}}_n$, which is defined as:

$$\begin{aligned}\hat{\mathcal{E}}_n = & \big\|\hat{C}_n(\mathbf{x}_{n0}, \mathbf{x}_{n0}) - \hat{\mathbf{y}}_n\big\|_2^2 + \lambda_1 \big\|\hat{\mathbf{w}}_n^*\big\|_2^2 \\ & + \sum_{s=1}^{S} \Big\| F_{ns} \hat{C}_n(\mathbf{x}_{ns}, \mathbf{x}_{ns}) \Big\|_2^2 \\ & + \sum_{t=1}^{T} \Big\| \gamma_t [\hat{\mathbf{w}}_n^* - \hat{\tilde{\mathbf{w}}}_n^*(t)] \Big\|_2^2\end{aligned} \ , \tag{5}$$

where the regularized factor for each context patch F_{ns} is defined as follows:

$$F_{ns} = \sqrt{\lambda_2} f_{ns} \ (s = 1, \cdots, S) \ , \tag{6}$$

where the importance facor f_{ns} to each patch s is introduced in detail in section 3.3.

Therefore, the solution to the optimization problem Eq. (5) can be calculated by setting the first derivative of $\hat{\mathbf{w}}_n^*$ to zero, i.e.:

$$\frac{\partial \hat{\mathcal{E}}_n}{\partial \hat{\mathbf{w}}_n^*} = 0 \ . \tag{7}$$

Since all the operations in Eq. (7) are performed in element-wise, as a result, a close-form solution to $\hat{\mathbf{w}}_n^*$ can be achieved by:

$$\hat{\mathbf{w}}_n^* = \frac{\hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}} \odot \hat{\mathbf{y}}_n + \sum_{t=1}^{T} \Big[{\gamma_t}^2 \hat{\tilde{\mathbf{w}}}_n(t) \Big]}{\hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}} \odot \hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}*} + \sum_{t=1}^{T} {\gamma_t}^2 + \lambda_1 + B_n} \ , \tag{8}$$

where the fraction operator, i.e., $\frac{*}{*}$, denotes element-wise division, and B_n is defined by (a detailed derivation is in Appendix 5):

$$B_n = \sum_{s=1}^{S} \Big({F_{ns}}^2 \hat{\mathbf{k}}^{\mathbf{x}_{ns}\mathbf{x}_{ns}} \odot \hat{\mathbf{k}}^{\mathbf{x}_{ns}\mathbf{x}_{ns}*} \Big) \ . \tag{9}$$

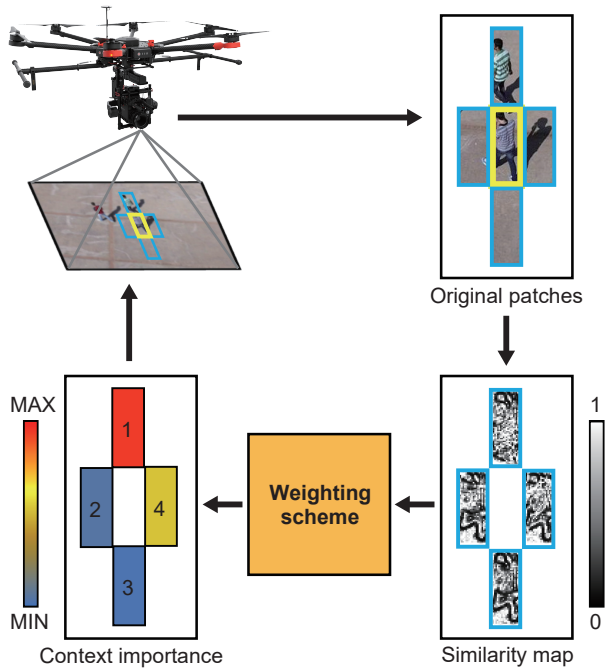

Fig. 3 Adaptive GMSD-based context analysis scheme. The similarity is utilized to compute the GMSD index of each context patch referring to object patch. Different importance of context patches is denoted by rectangles with different colors

3.3 Adaptive GMSD-based context analysis scheme

In most cases, the introduction of excessive context information undermines the discriminative power of the tracker, especially on the cluttered background. Therefore, the adaptive scheme measuring the importance of different patches referring to the object, i.e., the GMSD-based similarity index, is applied to utilize sample information and improve tracking accuracy.

The importance factor to each patch f_{ns} is calculated as follows:

$$f_{ns} = \frac{1}{C_{ns}} \exp(1 - GMSD_{ns}) \ , \tag{10}$$

where C_{ns} is the regularization term and $GMSD_{ns}$ is the context similarity referring to center patch.

In this work, the sample similarity is defined by GMSD [37] instead of assigning equal importance or using the Euclidean distance. The GMSD index is defined as:

$$GMSD = \sqrt{\frac{1}{P}\sum_{i=1}^{P}(GMS(i) - GMSM)^2} \ , \tag{11}$$

where GMS and $GMSM$ are represent as gradient magnitude similarity map at location i and gradient magnitude similarity mean to total pixels number P respectively, which are computed as follows:

$$GMS(i) = \frac{2\mathbf{m}_r(i)\mathbf{m}_d(i) + c}{\mathbf{m}_r^2(i) + \mathbf{m}_d^2(i) + c} \ , \tag{12}$$

and

$$GMSM = \frac{1}{P}\sum_{i=1}^{P} GMS(i) \ , \tag{13}$$

where $\mathbf{r}$ and $\mathbf{d}$ are the horizontal and vertical gradient images yielded by the convolving the reference and compared images with Prewitt filters $\mathbf{h}_x$ and $\mathbf{h}_y$ along horizontal and vertical directions. Besides, $\mathbf{m}_r(i)$ and $\mathbf{m}_d(i)$ are the gradient magnitudes of $\mathbf{r}$ and $\mathbf{d}$ at location i, which are computed as follows:

$$\begin{cases} \mathbf{m}_r(i) = \sqrt{(\mathbf{r} \otimes \mathbf{h}_x)^2(i) + (\mathbf{r} \otimes \mathbf{h}_y)^2(i)} \\ \mathbf{m}_d(i) = \sqrt{(\mathbf{d} \otimes \mathbf{h}_x)^2(i) + (\mathbf{d} \otimes \mathbf{h}_y)^2(i)} \end{cases} , \tag{14}$$

where $\otimes$ denotes the convolution operation.

Besides, for the context patches, learning at each frame may reduce the discriminative power of tracker, which is called an over-fitting phenomenon. Therefore, learning with adaptive GMSD-based analysis scheme updates at an equal interval of Δ_c. When the surrounding patches are not taken into account, context-aware regularization term λ_2 is set to zero, i.e., $B_n = 0$, to train the proposed framework Eq. (8) as follows:

$$\hat{\mathbf{w}}_n^* = \frac{\hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}} \odot \hat{\mathbf{y}}_n + \sum_{t=1}^{T}\left[\gamma_t{}^2 \hat{\tilde{\mathbf{w}}}_n(t)\right]}{\hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}} \odot \hat{\mathbf{k}}^{\mathbf{x}_{n0}\mathbf{x}_{n0}*} + \sum_{t=1}^{T}\gamma_t{}^2 + \lambda_1} \ . \tag{15}$$

Remark 2: In this work, the GMSD index can effectively measure the importance of each patch comparing to the object patch. As shown in Fig. 3, less importance will be assigned to the patch when it is more similar to the object. Thus, the proposed scheme can contribute to suppressing distractors adequately and construct the adaptive context restraints to ensure the robustness of the tracker. To avoid over-fitting, the number of context patches S is 4, and the interval of context patches Δ_c is set to 10.

3.4 Combination of multiple features

In this work, the features of object appearance, i.e., color, texture, and semantic information which respectively correspond to features of CN [31], HOG [3], and deep features extracted from CNN are selected to represent extracted patches.

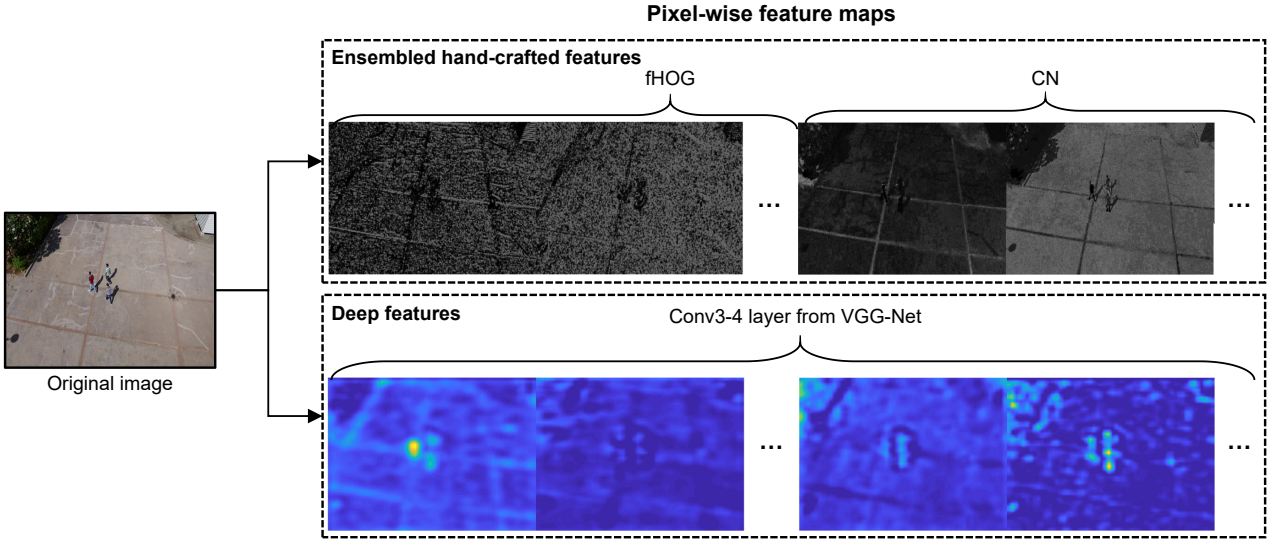

Fig. 4 Feature representations of the original image. Two different features are utilized to represent the original image, which include the ensembled hand-crafted features and convolutional feature extracted from VGG-Net

3.4.1 Object representation with ensembled hand-crafted features

In general, HOG computes on a dense grid of uniformly spaced cells in the patch and has the advantage against local geometric variance and photometric transformations. Besides, the integration of spatial sampling and local photometric normalization permits the body movement of tracking object to be ignored so that HOG can be robust to illumination variation.

In addition to texture characteristics, CN can be used to represent the object in terms of multiple color attributes. Compared to the original RGB expression, CN feature encoded the patches with the preselected set of 11 linguistic color labels to exploit the essential color information. It can tackle issues about object deformation and variation in shape through the perception of the object color.

The two features, i.e., HOG and CN, focus on the texture and color information of the patches; therefore, they can strengthen the comprehensive representation of the appearance model complementarily. For implementation, the two hand-crafted features are ensembled as one feature to encode the extracted center as well as the context patches.

3.4.2 Object representation with convolutional features

Compared to hand-crafted features, convolutional neural networks trained on the large-scale ImageNet dataset with the category-level label has the semantic-aware capability. Thus, the output of the convolutional layer with high-level encoding capability is employed after removing the fully-connected layers in this work. Moreover, resizing each feature map to a fixed larger size with bi-linear interpolation alleviates the issue of resolution degradation corresponding to the original patch.

Remark 3: Figure 4 shows the multiple features utilized in the proposed method. The fast version of HOG, i.e., fHOG [9], which has no information loss, is applied in this work. For CN, the mapping method proposed in [5] is employed to transform the RGB space into the color names space, which is an 11-dimensional color representation, i.e., black, blue, brown, gray, green, orange, pink, purple, red, white, and yellow. For deep features, VGG-Net [30] is used to extract the convolutional features in this work. Concerning the calculation complexity, the features extracted from the conv3-4 layer are only used to express the object.

3.5 Dynamic weighted filters training strategy

In this section, the multiple dynamic filters with time-attenuated factors are introduced in the model training stage to cope with arbitrary object appearance changes.

From the beginning, the new filters are selected as $\widetilde{\mathbf{w}}_n^*(1)$ to add into the pool at an interval Δ_f. Correspondingly, the remaining filters further from the current frame continue to be retained and passed backwards one after the other, i.e., $\widetilde{\mathbf{w}}_n^*(t-1)$ in the k-th frame is equal $\widetilde{\mathbf{w}}_n^*(t)$ in the $(k+\Delta_f)$-th frame. The maximum number of dynamic weighted filters is T, that is, the historical information in the latest $(\Delta_f \times T)$ frames are taken into account in the filter pool and the filters furthest from the current frame will be discarded. Besides, the penalty factor γ_t is assigned to different fil-

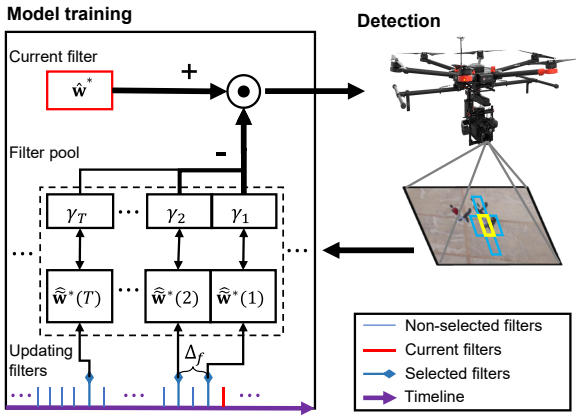

Fig. 5 Dynamic weighted filters training strategy. The filter pool updates at an interval Δ_f. According to the distance from the current frame, the penalty factor is assigned to different filters. The updating filters with blue diamond denotes selected filters in overall tracking process

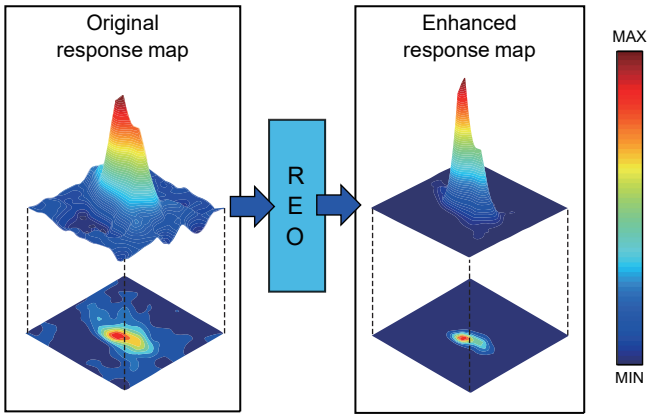

Fig. 6 Resolution enhancement operation. The enhancement of final response map corresponding to each correlator leads to less noise and sharper peak

ters $\widetilde{\mathbf{w}}_n^*(t)$ with regard to the time-attenuated influence, which defined as follows:

$$\gamma_t = \frac{C_f}{2^t} \ , \tag{16}$$

where C_f is the multi-filter regularization term, and t is the ordinal number of the t-th filter in the filter pool close to the current frame filter.

Figure 5 shows that multi-filters are added as dynamic restraints to guarantee accurate detection in a continuous video sequence.

Remark 4: Generally, most trackers ignore the historical information of the filters, and the update of the model exclusively depends on the appearance information changes. In the complex environment, the model cooperated with the filter pool will maintain similarity with the past information to achieve robustness. Unlike the proposed method in [21], multiple dynamic filters with attenuated factors are taken into account in this framework, and a closed-form solution is obtained to ensure accurate model updating. In this work, $T = 5$ weighted filters are training as restraints, and the update interval for multi-filters Δ_f is set to 2, and the multi-filter regularization term C_f is set 1.

3.6 Resolution enhancement operation

To increase model sensitivity, a simple yet effective strategy to enhance different correlators' responses is designed in this work to achieve better tracking accuracy.

The response map $\mathcal{R}_{\mathrm{HC}}$ and $\mathcal{R}_{\mathrm{CNN}}$ can be separately enhanced by following resolution enhancement operator (REO) which is defined as:

$$\mathcal{S}_{ij} = \sqrt{\sigma(\mathcal{R}_{ij}^2) \cdot \sum_{i,j=1}^{N} \mathcal{R}_{ij}^2} \ , \tag{17}$$

where $\mathcal{R}_{ij}$ is the original element in i_{th} row, j_{th} column of response map $\mathcal{R}$, and $\mathcal{S}_{ij}$ is the element in i_{th} row, j_{th} column of enhanced response map $\mathcal{S}$ after operation, and the enhancement function $\sigma(\cdot)$ is defined as:

$$\sigma(\mathcal{R}_{ij}^2) = \frac{b^{\mathcal{R}_{ij}^2}}{\sum\limits_{i,j=1}^{N} b^{\mathcal{R}_{ij}^2}} \ , \tag{18}$$

where b is the base coefficient.

Figure 6 shows that the peaks on both response maps after resolution enhancement can assist in realizing accurate and robust object detection.

For the consequent response fusion. the enhanced response map $\mathcal{S}_{\mathrm{HC}}$ and $\mathcal{S}_{\mathrm{CNN}}$ are combined to employ their complementary ability to localize object. Since the object appearance is changing over time, to improve the robustness of multi-kernelized correlators, an online adaptation strategy is used. At frame k, the model is updated with training rate η as follows:

$$\begin{aligned} \hat{\mathbf{w}}_k^{(\mathrm{model})} &= (1-\eta)\hat{\mathbf{w}}_{k-1} + \eta\hat{\mathbf{w}}_k \\ \hat{\mathbf{k}}_k^{\mathbf{xx}(\mathrm{model})} &= (1-\eta)\hat{\mathbf{k}}_{k-1}^{\mathbf{xx}} + \eta\hat{\mathbf{k}}_k^{\mathbf{xx}} \end{aligned} \ , \tag{19}$$

where subscript k and $k-1$ denote current frame and the model learned in the last update, respectively.

Remark 5: In order to adapt to appearance change, the learning rate η is set to 1×10^{-2} and kept the same in the evaluation stage. Details of the MKCT tracker can be seen in Algorithm 1.

Algorithm 1: MKCT tracker

Input: Frames of the video sequence: $I_1, \cdots, I_K$.
The interval for context learning: Δ_c.
The number of context pateches: S.
The interval for updating multi-filters: Δ_f.
Current dynamic filter pool: $\hat{\tilde{\mathbf{w}}}(1), \cdots, \hat{\tilde{\mathbf{w}}}(T)$.
Initialize the MKCT in the first frame.
Output: Predicted location in Frame # k.

for $k = 2$ *to end* **do**
  Extract the object patch in Frame # k from center location of the object in Frame # $k-1$
  Represent $\mathbf{x}_{n0}$ using ensembled hand-crafted (fHOG + CN) and deep features
  if mod $(k, I_f) == 0$ **then**
    foreach current filters $\hat{\tilde{\mathbf{w}}}(t), (t > 2)$ **do**
      Update multi-filters $\hat{\tilde{\mathbf{w}}}(t) = \hat{\tilde{\mathbf{w}}}(t-1)$
    Add latest updated model $\hat{\tilde{\mathbf{w}}}(1) = \hat{\mathbf{w}}_{k-1}^{(\text{model})}$
    Calculate the kernelized correlation $\hat{C}(\mathbf{x}_{n0}, \mathbf{x}_{n0})$ output using Eq. (2)
  else
    Calculate the kernelized correlation $\hat{C}(\mathbf{x}_{n0}, \mathbf{x}_{n0})$ output using Eq. (2)
  end
  Fuse different response maps with resolution enhancement operation by Eq. (17)
  Predict the object location in Frame # k by searching the maximum on the final fused map
  if mod $(k, I_c) == 0$ **then**
    Extract four context patches around the object
    foreach context patch $\mathbf{x}_{ns}$ **do**
      Represent extracted patches using ensembled hand-crafted (fHOG + CN) features and calculate GMSD-based similarity factor f_s by Eq. (10)
    Learn new object appearance by Eq. (8)
    Update the model $\hat{\mathbf{w}}_k^{(\text{model})}$ by Eq. (19)
  else
    Learn new object appearance by Eq. (8)
    Update the model $\hat{\mathbf{w}}_k^{(\text{model})}$ by Eq. (19)
  end
end

4 Experiment and Evaluation

4.1 Implementation details

For the proposed MKCT tracker, it is implemented in MATLAB 2017b. The same computer generates all the experimental results with Intel i7 processor (3.70 GHz), 32 GB RAM, and NVIDIA Quadro 2000 GPU. The MatConvNet toolbox is used for extracting the output of the conv3-4 layer from the VGG-Net [30].

Following basic settings in KCC [32], the search window of both center and context patches are extracted according to the object size whose area threshold is 50 pixels $\times$ 50 pixels and then processed with a Hanning window. The ensembled hand-crafted features containing fHOG and CN, as well as deep features extracted from the conv3-4 output of VGG-Net are used to represent the patch separately. For the response map corresponding to each kernelized correlator, the result is enhanced by REO. Besides, Table 1 shows additional parameters utilized in this work. All parameters are fixed in the following experiments.

Remark 6: The source code and related UAV tracking video of the proposed MKCT tracker can be found at `https://github.com/vision4robotics/MKCT-tracker` and `https://youtu.be/duSk2XMf5O4`. Besides, Fig. 14 shows examples of UAV tracking results.

4.2 Evaluation criteria

On the whole, precision and success are used to evaluate the performance of the tracking approach. The center location error (CLE) and the intersection over union (IoU), which are based on the one-pass evaluation protocol, can be employed to assess the two aspects above.

On the one hand, the CLE is defined as the distance of the bounding box center between the tracker and ground truth in pixel-wise, which is used to display the precision plot (PP). The threshold at 20 pixels is commonly used to rank the precision of each tracker.

On the other hand, the IoU of the tracker bounding box and ground truth bounding box can be computed to display the success plot (SP). Generally, the area under the curve (AUC) of SPs is selected to rank the success rate of each tracker.

4.3 Evaluation with state-of-the-art trackers

Based on the mentioned criteria, the performances of the proposed tracker can be demonstrated. The MKCT tracker is extensively evaluated on 100 UAV tracking sequences as challenging UAV image dataset with other 15 state-of-the-art trackers based on hand-crafted features, i.e., STRCF [21], MEEM [39], DCF_CA, Staple_CA, SAMF_CA [27], MCCT-H [33], MUSTER [15],

Table 1 Main parameters in the MKCT tracker

Paramter	Value
Update interval for multi-filters Δ_f	2
Number of multi-filters T	5
Number of context patches S	4
Interval for context learning Δ_c	10
Base of REO b	3
Model regularization term λ_1	1×10^{-4}
Context-aware regularization term λ_2	1/256
Multi-filter regularization term C_f	1

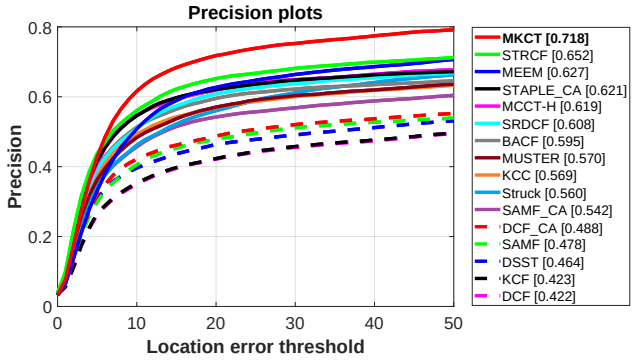

Fig. 7 PPs of MKCT tracker and 15 hand-crafted feature based trackers on the challenging UAV image dataset. MKCT tracker shows superiority in precision score, followed by STRCF and MEEM

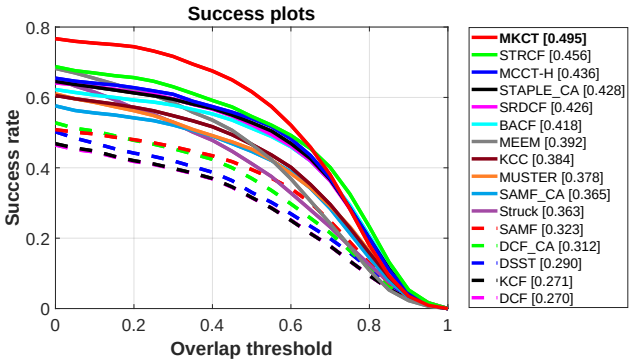

Fig. 8 SPs of MKCT tracker and 15 hand-crafted feature based trackers on the challenging UAV image dataset. MKCT tracker shows superiority in success rate, followed by STRCF and MCCT-H

SRDCF [6], BACF [18], KCC [32], Struck [12], SAMF [22], DCF, KCF [14], and DSST [4], and 8 trackers based deep learning, i.e., MCCT, CF2 [24], PTAV [8], CoKCF [41], DeepSTRCF, UDT, UDT+ [34], and MCPF [43].

Remark 7: Objective evaluations of the state-of-the-art tracking approaches are performed by utilizing the open-source codes as well as default parameters provided by the authors.

4.3.1 Evaluation with trackers based on hand-crafted features

Figure 7 and 8 show PPs and SPs of MKCT tracker and 15 trackers based on hand-crafted features. The plots demonstrate that the proposed tracking approach achieves the highest precision score (0.718) and the highest AUC score (0.495). The proposed MKCT tracker, which employs the convolutional feature, has favorably outperformed trackers using only hand-crafted features, thereby effectively ensuring the stability in UAV object tracking tasks.

4.3.2 Evaluation with trackers based on deep learning

The proposed MKCT tracker is also compared to 8 state-of-the-art trackers based on deep learning on the same challenging UAV image dataset. These deep-based methods include trackers using end-to-end deep neural network architectures, i.e., UDT, UDT+, and PTAV, or integrating convolutional features from a pre-trained deep network, i.e., CoKCF, CF2, MCPF, MCCT, DeepSTRCF, and MKCT. As shown in Table 2, the MKCT tracker (0.718) has an advantage of 3.6% over the second and third best tracker UDT+ (0.693) and PTAV (0.693) in precision, as well as an advantage of 0.8% and 2.9% over the DeepSTRCF (0.491) and MCCT (0.481) in AUC score.

4.3.3 Evaluation on 12 attributes and tracking speed

The trackers with the top 10 precision scores on the challenging UAV image dataset, i.e., MKCT, UDT+, PTAV, DeepSTRCF, MCCT, MCPF, STRCF, CF2, CoKCF, and MEEM, are selected for attribute and tracking speed evaluation.

Attribute-based comparison: The tracking attributes can be classified as follows: aspect ratio change (ARC), background clutter (BC), camera motion (CM), fast motion (FM), full occlusion (FOC), illumination variation (IV), low resolution (LR), out-of-view (OV), partial occlusion (POC), scale variation (SV), similar object (SOB) and viewpoint change (VC) [26]. Figure 12 and 13 show the precision and success plots with respect to different attributes. For precision, the MKCT tracker favorably outperforms other state-of-the-art trackers in all attributes except for FM. In terms of success rate, MKCT exhibited the best performance except for FM, OV, and FOC. In summary, the quantitative attribute-based experiments show that the MKCT tracker ranks

Table 2 Overall performance comparisons of MKCT tracker and 8 state-of-the-art trackers based on deep learning on the challenging UAV image dataset. The **red**, **green**, and **blue** fonts indicate the first, second and third place in terms of precision and success rate, respectively

Tracker	Published in	Prec.	Succ.
UDT	CVPR2019	0.596	0.428
UDT+	CVPR2019	**0.693**	0.475
PTAV	ICCV2017	**0.693**	0.475
CoKCF	PR2017	0.634	0.396
CF2	ICCV2015	0.635	0.431
MCPF	TPAMI2017	0.674	0.437
MCCT	CVPR2018	0.679	**0.481**
DeepSTRCF	CVPR2018	0.686	**0.491**
MKCT	**Ours**	**0.718**	**0.495**

*No.*1 in general among all trackers.

Tracking speed comparison: These 10 trackers are further evaluated in terms of the tracking speed. As shown in Table 3, the UDT+ achieves the fastest tracking speed, followed by STRCF based on hand-crafted features and the proposed MKCT tracker. Although UDT+ uses the fewest time to process every frame, its overall and attribute-based tracking performance are inferior to the MKCT tracker, which is implemented in MATLAB without engineering optimizations.

4.4 Extensive evaluations of the proposed method

In this section, quantitative and qualitative experiments are conducted to verify the effectiveness of the MKCT tracker.

4.4.1 Ablation studies

The proposed tracking approach with different modules are evaluated on the challenging UAV image dataset. First, KCC [32] is considered as a special case of MKCT with $N = 1$, $S = 0$, and $T = 0$. Second, MKCT-H is the proposed method utilized the hand-crafted feature only. Third, eMKCT is a variant of MKCT whose features and weighted filters selection are the same as those adopted by MKCT but with fixed context importance. Finally, MKCT-NR is the MKCT without the final enhancement operation before the response fusion.

As shown in Fig. 9, MKCT-NR outperforms eMKCT, MKCT-HC, and KCC with large margins in overall performance. The novel resolution enhancement function of MKCT improves the performance with the average precision score of 2.9% and the average AUC score of 2.3% to MKCT-NR, and significantly outperforms eMKCT and MKCT-HC by 4.4% and 7.4% in precision score and 4.3% and 5.2% in AUC score.

Table 3 Millisecond per frame (MSPF) and average frame per second (FPS) of top 10 trackers on the challenging UAV image dataset. The **red**, **green**, and **blue** fonts indicate the first, second, and third place

	Trackers	MSPF	FPS
Hand-crafted features	MEEM	126.6	7.9
	STRCF	**49.5**	**20.2**
Deep learning	UDT+	**25.5**	**39.2**
	PTAV	212.8	4.7
	CoKCF	416.7	2.4
	CF2	161.3	6.2
	MCPF	1724.1	0.6
	MCCT	1428.6	0.7
	DeepSTRCF	217.4	4.6
	MKCT	**106.4**	**9.4**

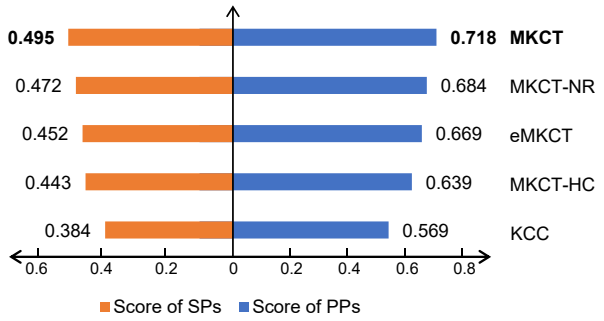

Fig. 9 Ablation study of the proposed MKCT tracker on the challenging UAV image dataset. The overall results demonstrate the effective improvement of each module

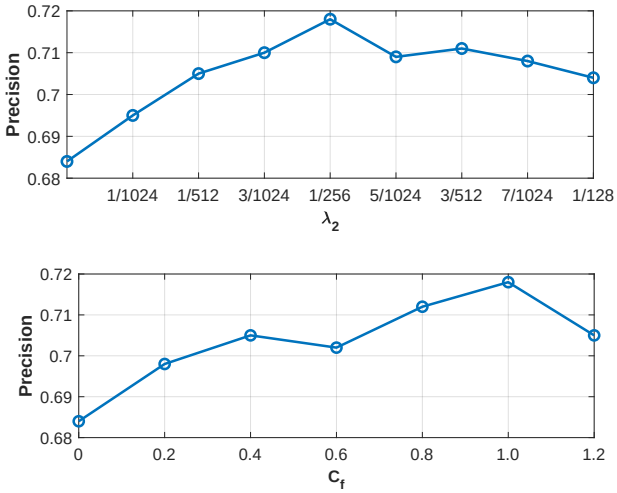

Fig. 10 Effect of context-aware regularization term λ_2 and multi-filter regularization term C_f on the challenging UAV image dataset. When λ_2 and C_f are set more than 0, both key parameters contribute to the overall performance of MKCT tracker

4.4.2 Key parameter analysis

To validate the effectiveness of the regularization terms on the overall performance, different λ_2 and C_f are further analyzed on the same challenging UAV image dataset. During the analysis of the parameters, λ_2 ranges from 0 to 1/128 with a step size of 1/1024, and C_f ranges from 0 to 1.2 with a step size of 0.2.

The regularization parameter λ_2 determines the degree of GMSD-based constraints imposed by the adaptive surrounding context: the lower λ_2 is set, the less attention is paid to the possible inference caused by the cluttered contexts. Likewise, the regularization parameter C_f determines the degree of adaptive temporal constraints imposed by dynamic weighted filters in the training stage: the lower C_f is set, the less focus is kept on the past filters. MKCT with $\lambda_2 = 0$ and $C_f = 0$ equals the filters trained without the consideration of changing context information and dynamic historical

information, respectively. Figure 10 shows when C_f is fixed and λ_2 is set over 0, MKCT effectively improves the precision score and reach the peak (0.718) at $\lambda_2 = 1/256$. Similarly, when λ_2 is fixed and C_f is set over 0, MKCT maintains a advanced precision level and reach the highest score (0.718) at $C_f = 1$. Thus, $\lambda_2 = 1/256$ and $C_f = 1$ are chosen to demonstrate state-of-the-art performance on the challenging UAV image dataset.

4.4.3 Time analysis

To analyze the time cost, one of the challenging UAV image dataset, i.e., *group1_1* with 445 frames, is selected. Table 4 shows that each stage of MKCT is operated within acceptable time, and the proposed tracker spends a total of 45.41s with speed at 9.8 FPS in the sequence, which proves its efficiency. Due to the introduction of multiple features and adaptive context analysis strategy, the MKCT tracker reaches an average speed at 9.4 FPS on the challenging UAV image dataset.

Remark 8: The MKCT tracker is currently implemented on MATLAB without optimization, and its efficient application can be achieved with the help of parallel computation and optimization strategy on GPU. In the real-world UAV tracking tasks, the multirotor can carry a larger payload with high-performance GPU and CPU, which will significantly improve the efficiency of the tracker.

4.5 Limitations and future work

Though the MKCT tracker has performed better against other state-of-the-art trackers in the overall evaluation, it still has certain limitations during the UAV tracking. Figure 12 and 13 show that the MKCT tracker has obtained inferior precision performance in FM and lower success rate in the FM, OV, and FOC attributes compared to other trackers.

Therefore, two challenging sequences in Fig. 11, in which MKCT fails to track the object, are used to investigate the underlying causes and discuss future improvements. In the sequence *car7*, the noisy background contains a similar object and interfering trees, which cause the proposed method to drift from the object in Frame # 75. In such cases, MKCT can combine the motion information in [11] and the variable aspect ratio module introduced in [20] to solve these problems. In the sequence *person14_1*, the tracked person is initialized with a low-resolution bounding box. From Frame # 51 to # 65, the person moved quickly into the pavilion, resulting in full occlusion. Like most CF-based methods, MKCT can hardly detect the disappearance and reappearance of the tracked object due to the lack of re-detection modules such as those used in MEEM and TLD [17]. We strongly believe the MKCT can retrieve objects after returning to the field of view by adding a re-detection scheme, thereby further improving performance.

Table 4 Time cost of the proposed MKCT in each stage on the sequence *group1_1* with 445 frames

Stage	Time	Ratio
Object patch extraction & representation	26.22s	57.7%
Context patches extraction & representation	2.99s	6.6%
CF training with different kernels	8.16s	18.0%
Response generation & fusion	1.60s	3.5%
New location prediction	1.35s	3.0%
Other operations	5.09s	11.2%
Total time	45.41s	100.0%

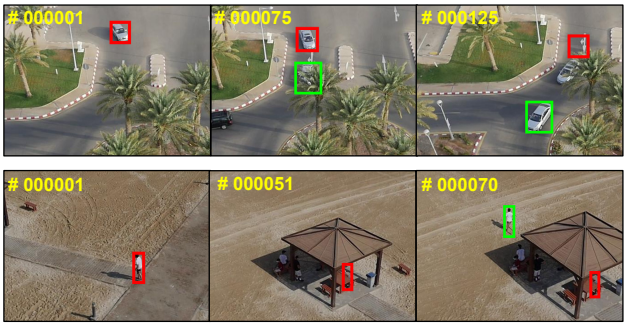

Fig. 11 Failure cases of the proposed method from sequences *car7* (first row) and *person14_1* (second row), where the **red** and **green** bounding boxes denote our tracking result and ground truth

5 Conclusion and outlook

In this work, the robust tracking approach, i.e., MKCT tracker, is presented for UAV to achieve state-of-the-art performance in complicated situations. Both the hand-crafted and convolutional features are utilized for training with dynamic weighted filters in kernel space. Thanks to the adaptive GMSD-based analysis scheme, the object and context patches are adequately exploited to reinforce the capability to distinguish the object from the background. Besides, a filter pool is constructed with dynamic weighting filters and updates at a frequency to make better use of historical information during the model update to ensure model robustness. Moreover, the REO is capable of sharpening the peak on the response map effectively over time. Consequently, extensive and in-depth experiments are conducted on 100 challenging UAV image sequences. The extensive experimental results show that the proposed MKCT tracker favorably outperforms 23 state-of-the-art track-

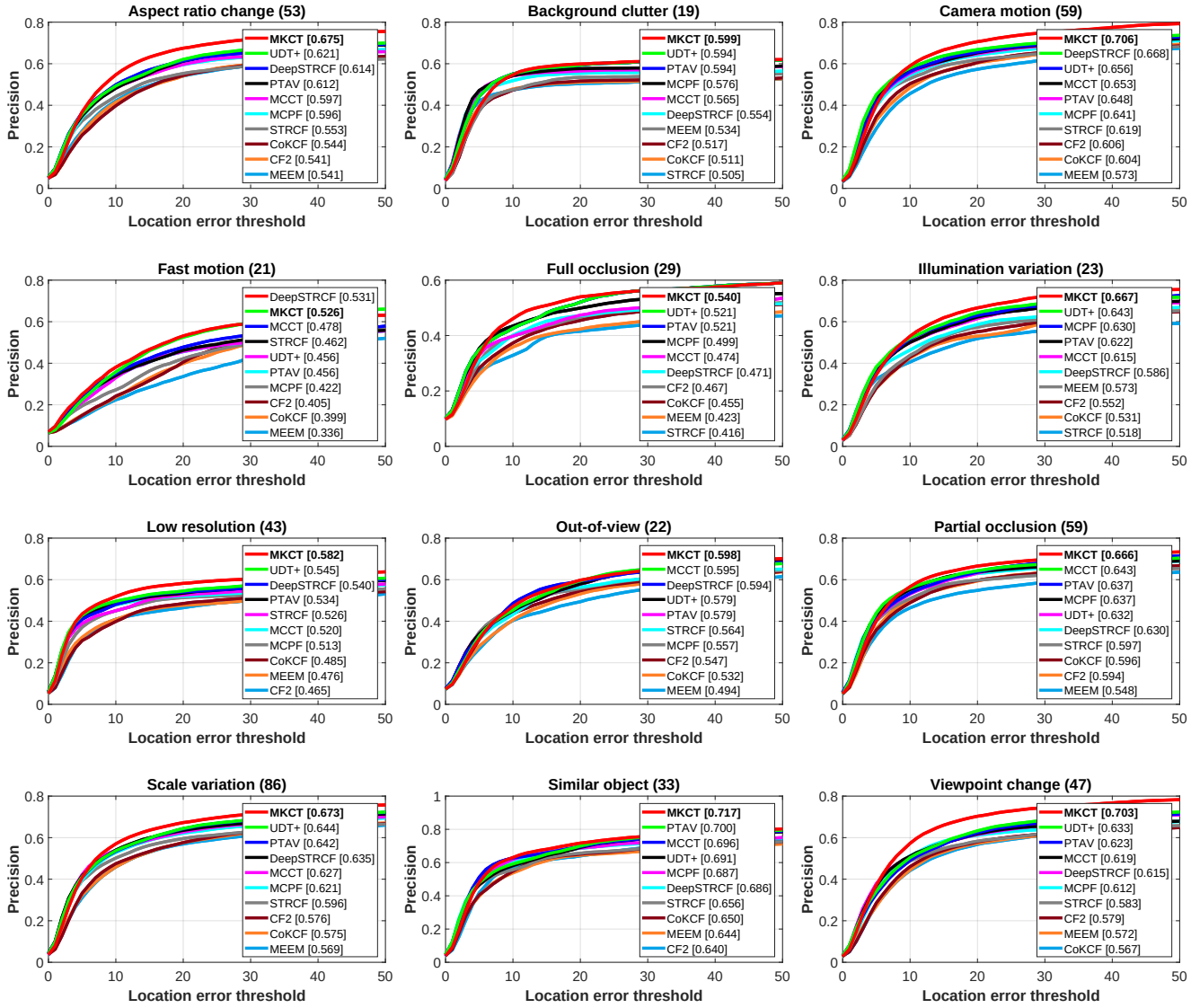

Fig. 12 PPs of top 10 trackers on different attributes with the challenging UAV image dataset

ers, including 15 trackers based on hand-crafted features and 8 trackers based on deep learning.

In the future, the proposed methods, including adaptive GMSD-based context analysis scheme and dynamic weighted filters strategy, can be generalized to other CF-based trackers like MCCT [33] and STRCF [21]. Besides, by incorporating our proposed strategies with more robust and lightweight convolutional features or similarity metrics, the performance of tracking methods for UAVs can be further improved. We believe that with our proposed method, a CF-based tracking framework can open the door to more extensive applications and more in-depth researches for UAV.

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

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest to this work.

Appendix

In this section, a more detailed derivation from Eq. (5) to Eq. (8) is presented.

Because all operations in the Fourier domain are performed element-wise, each element of $\hat{\mathbf{w}}_n^*$ (indexed by u) can be solved independently, and the Eq. (5) can be decomposed

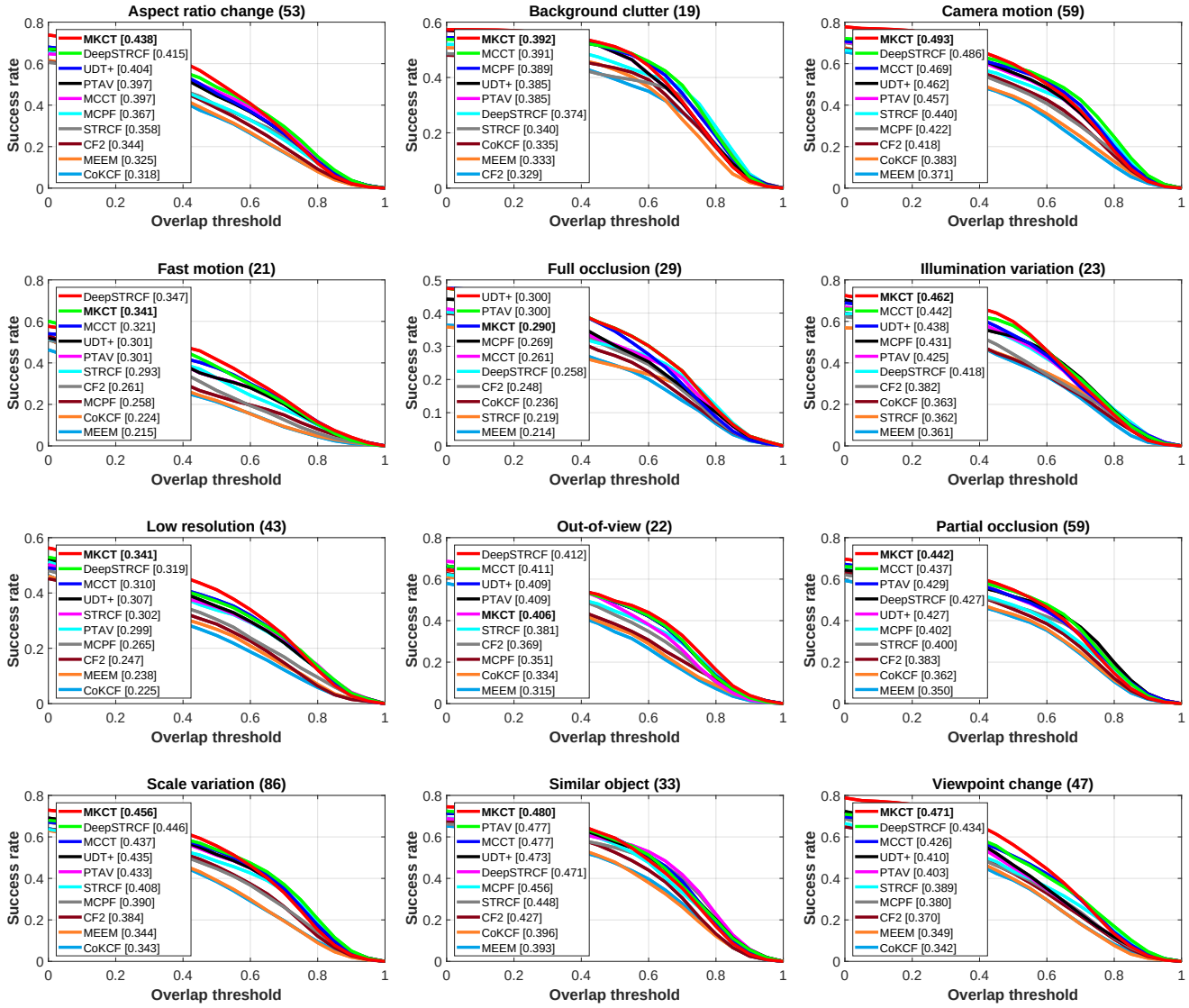

Fig. 13 SPs of top 10 trackers on different attributes with the challenging UAV image dataset

as the subploblem $\hat{\mathcal{E}}_{nu}$, which is defined as follows:

$$\begin{aligned}\hat{\mathcal{E}}_{nu} =& \left\|\hat{K}_u^{n0}\hat{w}_{nu}^* - \hat{y}_{nu}\right\|_2^2 + \lambda_1\left\|\hat{w}_{nu}^*\right\|_2^2 \\ &+ \sum_{s=1}^{S}\left\|F_{ns}\hat{K}_u^{ns}\hat{w}_{nu}^*\right\|_2^2 \\ &+ \sum_{t=1}^{T}\left\|\gamma_t\left[\hat{w}_{nu}^* - \hat{\tilde{w}}_{nu}^*(t)\right]\right\|_2^2\end{aligned}, \tag{20}$$

where $\hat{K}_u^{n0} = \hat{k}_u^{\mathbf{x}_{n0}\mathbf{x}_{n0}}$ and $\hat{K}_u^{ns} = \hat{k}_u^{\mathbf{x}_{ns}\mathbf{x}_{ns}}$ are used to simplify the denotation. Then, Eq. (20) can be expanded according to the property of the vector operation, that is equivalent to

$$\begin{cases} A_1 = \left\|\hat{K}_u^{n0}\hat{w}_{nu}^* - \hat{y}_{nu}\right\|_2^2 \\ \quad = \hat{K}_u^{n0*}\hat{w}_{nu}\hat{K}_u^{n0}\hat{w}_{nu}^* + \hat{y}_{nu}^*\hat{y}_{nu} \\ \quad - \hat{K}_u^{n0}\hat{w}_{nu}^*\hat{y}_{nu}^* - \hat{K}_u^{n0*}\hat{w}_{nu}\hat{y}_{nu} \\ A_2 = \sum_{t=1}^{T}\left\|\gamma_t\left[\hat{w}_{nu}^* - \hat{\tilde{w}}_{nu}^*(t)\right]\right\|_2^2 \\ \quad = \sum_{t=1}^{T}\left\{{\gamma_t}^2\left[\hat{w}_{nu}\hat{w}_{nu}^* + \hat{\tilde{w}}_{nu}(t)\hat{\tilde{w}}_{nu}^*(t)\right.\right. \\ \quad \left.\left. - \hat{w}_{nu}\hat{\tilde{w}}_{nu}^*(t) - \hat{w}_{nu}^*\hat{\tilde{w}}_{nu}(t)\right]\right\} \\ A_3 = \lambda_1\left\|\hat{w}_{nu}^*\right\|_2^2 = \lambda_1\hat{w}_{nu}\hat{w}_{nu}^* \\ A_4 = \sum_{s=1}^{S}\left\|F_{ns}\hat{K}_u^{ns}\hat{w}_{nu}^*\right\|_2^2 \\ \quad = \sum_{s=1}^{S}\left({F_{ns}}^2\hat{K}_u^{ns*}\hat{w}_{nu}\hat{K}_u^{ns}\hat{w}_{nu}^*\right) \end{cases}. \tag{21}$$

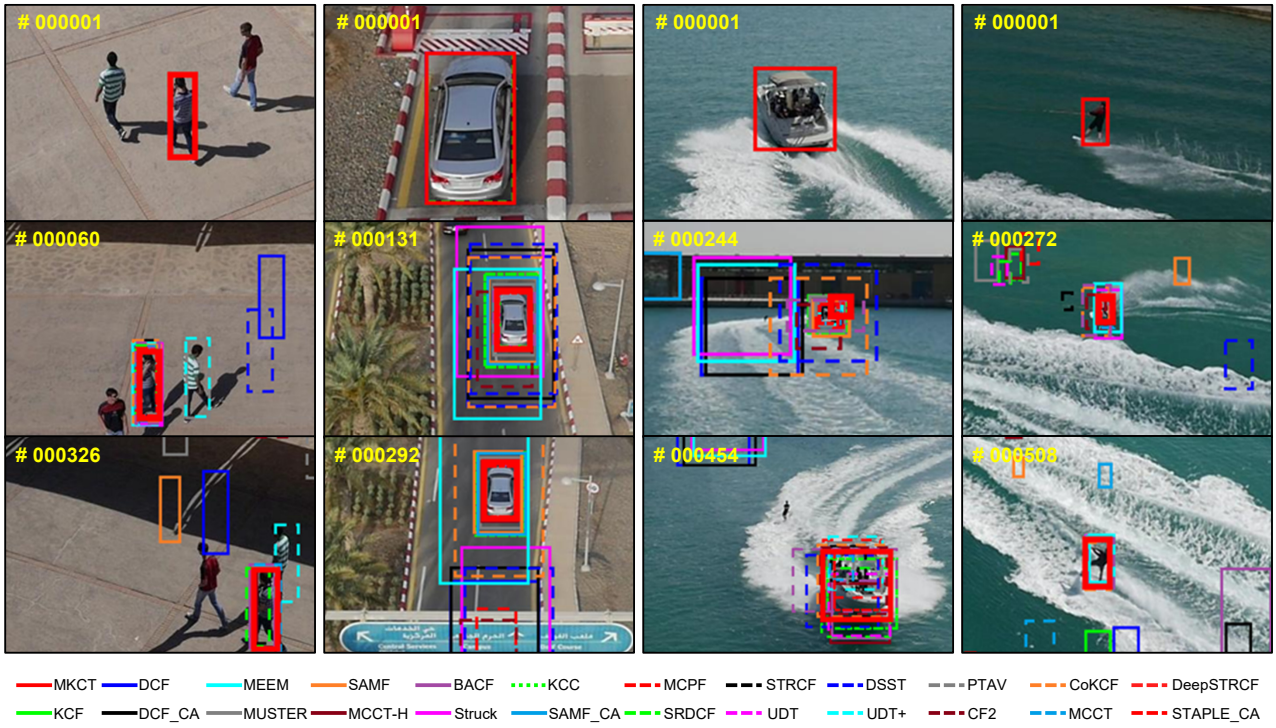

Fig. 14 Examples of UAV tracking results. The first, second, third and fourth column show the *group1_3*, *car9*, *boat9*, and *wakeboard8* image sequences

Therefore, the solution to the optimization target can be calculated by setting the first derivative of $\hat{w}^*_{nu}$ to zero, i.e.:

$$\frac{\partial \hat{\mathcal{E}}_{nu}}{\partial \hat{w}^*_{nu}} = \frac{\partial A_1}{\partial \hat{w}^*_{nu}} + \frac{\partial A_2}{\partial \hat{w}^*_{nu}} + \frac{\partial A_3}{\partial \hat{w}^*_{nu}} + \frac{\partial A_4}{\partial \hat{w}^*_{nu}} = 0 \ . \tag{22}$$

Hence, Eq. (22) can be reformulated as follows:

$$\begin{aligned} &\left[\hat{K}_u^{n0*}\hat{K}_u^{n0} + \sum_{t=1}^{T}{\gamma_t}^2 + \lambda_1 + \sum_{s=1}^{S}\left({F_{ns}}^2\hat{K}_u^{ns*}\hat{K}_u^{ns}\right)\right]\hat{w}_{nu} \\ &= \hat{K}_u^{n0}\hat{y}^*_{nu} + \sum_{t=1}^{T}\left[{\gamma_t}^2\hat{\tilde{w}}_{nu}(t)\right] \end{aligned} \ . \tag{23}$$

A closed-form solution to $\hat{w}^*_{nu}$ can be obtained:

$$\hat{w}^*_{nu} = \frac{\hat{K}_u^{n0}\hat{y}^*_{nu} + \sum_{t=1}^{T}\left[{\gamma_t}^2\hat{\tilde{w}}_{nu}(t)\right]}{\hat{K}_u^{n0}\hat{K}_u^{n0*} + \sum_{t=1}^{T}{\gamma_t}^2 + \lambda_1 + \sum_{s=1}^{S}\left({F_{ns}}^2\hat{K}_u^{ns}\hat{K}_u^{ns*}\right)} \ . \tag{24}$$

which is the sub-solution of $\hat{\mathbf{w}}^*_n$ in Eq. (8). ■

References

1. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PH (2016) Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1401–1409
2. Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2544–2550
3. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol 1, pp 886–893
4. Danelljan M, Häger G, Khan F, Felsberg M (2014) Accurate scale estimation for robust visual tracking. In: The British Machine Vision Conference (BMVC)
5. Danelljan M, Shahbaz Khan F, Felsberg M, Van de Weijer J (2014) Adaptive color attributes for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1090–1097
6. Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 4310–4318
7. Danelljan M, Robinson A, Khan FS, Felsberg M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 472–488
8. Fan H, Ling H (2017) Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 5486–5494
9. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence 32(9):1627–1645

10. Fu C, Duan R, Kayacan E (2019) Visual tracking with online structural similarity-based weighted multiple instance learning. Information Sciences 481:292–310
11. Gladh S, Danelljan M, Khan FS, Felsberg M (2016) Deep motion features for visual tracking. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp 1243–1248
12. Hare S, Golodetz S, Saffari A, Vineet V, Cheng M, Hicks SL, Torr PHS (2016) Struck: Structured Output Tracking with Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(10):2096–2109
13. Henriques JF, Caseiro R, Martins P, Batista JP (2012) Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In: Proceedings of the European Conference on Computer Vision (ECCV)
14. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3):583–596
15. Hong Z, Chen Z, Wang C, Mei X, Prokhorov D, Tao D (2015) Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 749–758
16. Hore A, Ziou D (2010) Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition, pp 2366–2369
17. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7):1409–1422
18. Kiani Galoogahi H, Fagg A, Lucey S (2017) Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1135–1143
19. Leichter I (2012) Mean shift trackers with cross-bin metrics. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(4):695–706
20. Li F, Yao Y, Li P, Zhang D, Zuo W, Yang MH (2017) Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2001–2009
21. Li F, Tian C, Zuo W, Zhang L, Yang MH (2018) Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4904–4913
22. Li Y, Zhu J (2014) A scale adaptive kernel correlation filter tracker with feature integration. In: ECCV workshop, pp 254–265
23. Lin S, Garratt MA, Lambert AJ (2017) Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment. Autonomous Robots 41(4):881–901
24. Ma C, Huang JB, Yang X, Yang MH (2015) Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 3074–3082
25. Ma K, Yeganeh H, Zeng K, Wang Z (2015) High dynamic range image compression by optimizing tone mapped image quality index. IEEE Transactions on Image Processing 24(10):3086–3097
26. Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for uav tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 445–461
27. Mueller M, Smith N, Ghanem B (2017) Context-aware correlation filter tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1396–1404
28. Olivares-Mendez M, Fu C, Ludivig P, Bissyandé T, Kannan S, Zurad M, Annaiyan A, Voos H, Campoy P (2015) Towards an autonomous vision-based unmanned aerial system against wildlife poachers. Sensors 15(12):31362–31391
29. Pednekar GV, Udupa JK, McLaughlin DJ, Wu X, Tong Y, Simone CB, Camaratta J, Torigian DA (2018) Image quality and segmentation. In: Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling, vol 10576, p 105762N
30. Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR)
31. Van De Weijer J, Schmid C, Verbeek J, Larlus D (2009) Learning color names for real-world applications. IEEE Transactions on Image Processing 18(7):1512–1523
32. Wang C, Zhang L, Xie L, Yuan J (2018) Kernel cross-correlator. In: AAAI Conference on Artificial Intelligence (AAAI)
33. Wang N, Zhou W, Tian Q, Hong R, Wang M, Li H (2018) Multi-cue correlation filters for robust visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4844–4853
34. Wang N, Song Y, Ma C, Zhou W, Liu W, Li H (2019) Unsupervised deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1308–1317
35. Wen D, Han H, Jain AK (2015) Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security 10(4):746–761
36. Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9):1834–1848
37. Xue W, Zhang L, Mou X, Bovik AC (2014) Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. IEEE Transactions on Image Processing 23(2):684–695
38. Yi S, Jiang N, Feng B, Wang X, Liu W (2016) Online similarity learning for visual tracking. Information Sciences 364:33–50
39. Zhang J, Ma S, Sclaroff S (2014) MEEM: robust tracking via multiple experts using entropy minimization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 188–203
40. Zhang K, Song H (2013) Real-time visual tracking via online weighted multiple instance learning. Pattern Recognition 46(1):397–411
41. Zhang L, Suganthan PN (2017) Robust visual tracking via co-trained Kernelized correlation filters. Pattern Recognition 69:82–93
42. Zhang L, Zhang L, Mou X, Zhang D (2011) FSIM: A feature similarity index for image quality assessment. IEEE transactions on Image Processing 20(8):2378–2386
43. Zhang T, Xu C, Yang MH (2017) Multi-task correlation particle filter for robust object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4335–4343