IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 28, 2019, accepted June 7, 2019, date of publication June 13, 2019, date of current version July 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922703

Part-Based Background-Aware Tracking for UAV With Convolutional Features

CHANGHONG FU[1], YINQIANG ZHANG[2], ZIYUAN HUANG[3], RAN DUAN[4], AND ZONGWU XIE[5]

[1]School of Mechanical Engineering, Tongji University, Shanghai 201804, China
[2]Department of Mechanical Engineering, Technical University of Munich, 80333 Munich, Germany
[3]School of Automotive Studies, Tongji University, Shanghai 201804, China
[4]Adaptive Robotic Controls Lab, Hong Kong Polytechnic University, Hong Kong
[5]State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150080, China

Corresponding authors: Changhong Fu (changhongfu@tongji.edu.cn) and Zongwu Xie (xiezongwu@hit.edu.cn)

This work was supported by the State Key Laboratory of Robotics and System (HIT) under Grant SKLRS-2018-KF-03.

ABSTRACT In recent years, visual tracking is a challenging task in UAV applications. The standard correlation filter (CF) has been extensively applied for UAV object tracking. However, the CF-based tracker severely suffers from boundary effects and cannot effectively cope with object occlusion, which results in suboptimal performance. Besides, it is still a tough task to obtain an appearance model precisely with hand-crafted features. In this paper, a novel part-based tracker is proposed for the UAV. With successive cropping operations, the tracking object is separated into several parts. More specially, the background-aware correlation filters with different cropping matrices are applied. To estimate the translation and scale variation of the tracking object, a structured comparison, and a Bayesian inference approach are proposed, which jointly achieve a coarse-to-fine strategy. Moreover, an adaptive mechanism is used to update the local appearance model of each part with a Gaussian process regression method. To construct a better appearance model, features extracted from the convolutional neural network are utilized instead of hand-crafted features. Through extensive experiments, the proposed tracker reaches competitive performance on 123 challenging UAV image sequences and outperforms other 20 popular state-of-the-art visual trackers in terms of overall performance and different challenging attributes.

INDEX TERMS Visual object tracking, unmanned aerial vehicle (UAV), convolutional neural network, background-aware correlation filter, part-based strategy, Gaussian process regression.

I. INTRODUCTION

Visual object tracking is a pivotal problem for unmanned aerial vehicle (UAV) applications. In recent years, various tracking methods have been developed to solve challenging problems, such as reconnaissance and surveillance [1], midair monitoring [2], and ship deck landing [3], autonomous chasing [4], infrastructure patrolling [5], pipeline inspection [6], air-to-air refuel [7] and precise landing [8]. Although a plethora of trackers is designed for UAV tracking applications, it is still a tough task to achieve robust tracking, especially in a complex environment. The key reason is the change of object appearance caused by deformation, illumination variation, partial or full occlusion, scale changes, motion blur, in-plane or out-of-plane rotation, low image resolution, cluttered background, fast motion, and camera motion. To address the issues mentioned above, an efficient and robust tracker with high accuracy is demanded inevitably.

In literature, UAV tracking methods can be separated into two types, namely discriminative and generative approaches. Generative trackers attempt to construct a robust appearance model or to learn it online using advanced machine learning techniques such as subspace learning [9], dictionary learning [10], sparse learning [11]. In contrast, discriminative approaches aim to train a binary classifier to distinguish the tracking object from its background, i.e, tracking-by-detection method. Their superior performance assists the trackers, such as compressive sensing [12], multiple instance learning [13] and structured output tracking with kernels [14], to dominate several visual tracking benchmarks [15], [16].

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

Recently, the correlation filter (CF)-based discriminative methods have been widely applied to challenging UAV tracking tasks with high-speed and promising performance. However, the standard CF tracker [17] confronts boundary effects and severe impacts of learning from circularly shifted samples from the foreground object. To tackle these issues, several background-aware trackers are proposed [18], [19]. Moreover, The spatial regularization, which enlarges the searching area and increases the tracking robustness, is incorporated into the CF-based tracking method [20]. However, the tracker with a holistic appearance model is prone to be dominated by occluded regions of the tracking object.

To solve the above problems, several CF-based trackers [21]–[25] have been combined with a part-based strategy for visual tracking applications. Also, the fusion of responses from multiple parts is still an intractable task. With particle filter, trackers [21] tackle this issue and achieve a promising performance. Also, particle filter plays a key roll in these works [26], [27], which achieve outstanding results. However, all the aforementioned CF-based trackers are based on hand-crafted features. In UAV applications, these features cannot maintain a comprehensive description of the tracking object because of its drastic deformation, occlusion, and rotation. To improve the representative ability of appearance model, several tracking methods based on convolutional features and CF have been developed, to name a few [28], [29]. These methods focus on integrating convolutional features from a fixed pre-trained deep network such as [30] and [31]. Moreover, some trackers [32]–[34] constructed with neural networks distinguish them from others with outstanding performances. The convolutional features strengthen these trackers against drastic changes in its appearance.

In this paper, a novel visual tracker for UAV is proposed. In summary, the main contributions of this presented work are listed as follows:

- A novel part-based visual tracker is proposed and applied for the UAV object tracking applications: Compared with the holistic appearance model, the part-based strategy endows the proposed tracker the ability against object partial and full occlusion, which is frequently confronted in aerial tracking.
- A new method with background-aware tracking for each part by successive cropping operation: The tracked parts are cropped by different cropping matrices and tracked by a background-aware discriminative tracking approach with convolutional features extracted by only conv3-4 layer in VGG-19 network [30].
- A novel approach with coarse-to-fine strategy is presented to estimate the location and scale changes of tracking object: With locations of parts, the structures of tracking object on two consecutive frames are compared for estimating the initial location and scale changes, and then a Bayesian inference framework is incorporated to refine the tracking results.
- A novel method is presented, which aims to update each local appearance model adaptively: The probabilistic relationship between the peak to sidelobe ratio (PSR) and smooth constraint of confidence maps (SCCM) is established. When the current PSR and SCCM of each part coincide with this relationship, the appearance model is updated.

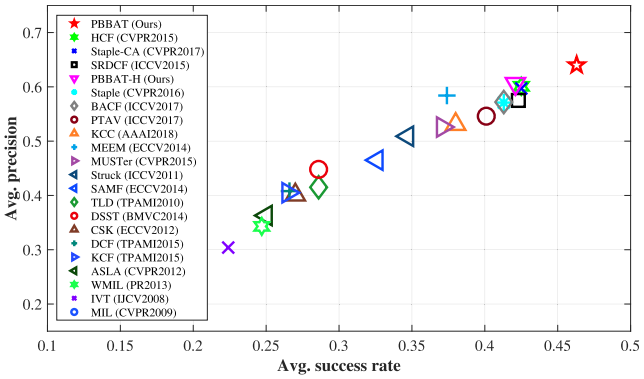


FIGURE 1. Overall precision and success rate of the proposed tracker and other state-of-the-art trackers on UAV123-10fps benchmark.

Extensive experiments show that the presented visual tracking approach achieves comparable performance on the computer with an i7-8700K processor (3.7GHz), 48GB RAM and NVIDIA Quadro P2000 GPU on UAV123-10fps benchmark, and outperforms 20 most popular state-of-the-art visual trackers in terms of robustness and accuracy, as shown in Fig. 1. To the best of our knowledge, it is the first time that this novel part-based background-aware visual tracker is presented, and applied for the UAV tracking applications in the literature.

The outline of this paper is organized as follows: Section II introduces the related tracking approaches with correlation filter, part-based mechanism, particle filter and trackers with convolutional features. Section III introduces the presented novel visual tracking algorithm, i.e., part-based background-aware visual tracker. Section IV presents the performance evaluations and comparisons with the most popular state-of-the-art visual trackers. Finally, the conclusion is given in Section V.

II. RELATED WORK

In this section, the state-of-the-art tracking approaches are introduced, which are closely related to this work.

A. CORRELATION FILTER BASED TRACKING

Recently, an autonomous vision-based system with kernelized correlation filter (KCF) [17] has been deployed on the UAV platform in [35] to track a maneuvering target efficiently. In the work [36], The CF tracker can achieve real-time, smooth and long-term object tracking from indoor to outdoor practical scenarios. Moreover, the KCF tracker is applied for the generation of an image patch confidence in literature [37], measuring the reliability of object tracking in the UAV applications. After the KCF tracker, in the work [38], an adaptive scale version of KCF, i.e., discriminative scale

space tracker (DSST) is proposed. In the presence of boundary effects and severe impacts of learning from circularly shifted samples of the foreground object, another approach proposed a spatial regularized CF-based tracker, i.e. spatially regularized correlation filters (SRDCF) [20]. The penalizing operation allows it to enrich the set of negative training samples without corrupting the positive ones. Another variant of CF-based trackers focuses on the background information around the tracking object. H. K. Galoogahi *et al.* propose a CF-based tracker with limited boundaries (CFLB) [19] and a background-aware CF-based tracker (BACF) [18] to crop with the boundary effects of circularly shifted patches with a cropping operation, i.e. masking matrix, which achieve outstanding performance. However, in UAV applications, visual tracking severely suffers from object occlusion. The aforementioned CF-based trackers cannot deal with it effectively.

B. TRACKING WITH PART-BASED STRATEGY

With the part-based strategy, multiple parts from the object can maintain traceable cues for tracking, which leads to a better tracking performance for UAV than approaches with a holistic appearance model. In the work [11], the object is divided into small patches by a regular grid, where the sparsity is adopted as the similarity metric. Combined with CF, the tracker presented in [21] utilize based on multiple correlation filters and the Bayesian inference framework. In literature [22], the reliable patch tracker, i.e., RPT, identifies and exploits the reliable patches that can be tracked effectively with a sequential Monte Carlo framework. A Hough voting-like scheme is applied to estimate the target state. The work in [23] has proposed a tracking method using dense belief propagation.

In visual tracking, the particle filter is an approach to construct the posterior probability density function of the state space recursively. In literature, Kwon and Lee [27] proposed a star-like appearance model, where a particle filter is employed to find the best state of the tracked object. Moreover, it builds the foreground and background models for segmentation to further refine the tracking results. Based on particle filtering, Ross et al. [9] employed the probabilistic principal component analysis (PCA) to represent target likelihoods by eigenbases. In [26], the observation likelihood is computed in a coarse-to-fine manner, which allows an efficient focus on more promising particles. Nevertheless, the features, used by the above trackers, are almost artificially designed. These hand-craft features cannot describe the appearance model precisely, especially in the visual tracking for UAV.

C. TRACKING WITH CONVOLUTIONAL FEATURES

Deep learning has pervaded many areas of computer vision. While these techniques have also been investigated for visual tracking. In literature [32], a stacked denoising autoencoder is trained offline to learn generic image features that are more robust against variations. In [33], the presented tracker uses a simple feed-forward network with no online training required. It can achieve real-time tracking performance and good accuracy. With convolutional features, The CF-based tracker with hierarchical convolutional features (HCF) [28] utilizes both early and last convolutional layers hierarchically to exploit semantic information of the tracking object with a competitive accuracy simultaneously. The features are the layer conv3-4, conv4-4 and conv5-4 extracted with VGG-19 network [30]. Also, DeepSDRCF tracker use convolutional features and obtain a promising improvement compared with its former version, i.e., SDRCF tracker [20]. In the parallel tracking and verifying framework (PTAV) [34], a tracker and a verifier are proposed. they work in a parallel way on two separate threads. This method enjoys both the high efficiency provided by the real-time tracker and the strong discriminative power by the verifier, which checks the tracking results upon the requests from the tracker. In UAV applications, convolutional features can provide a robust and quasi-invariant description of the tracking object, which has a demonstrable effect on the improvement of the tracking performance on a UAV platform.

III. PROPOSED TRACKING APPROACH

In this section, a thorough introduction of the tracking framework for UAV, i.e., PBBAT tracker, is presented.

A. OVERVIEW

The overall tracking framework of the proposed PBBAT tracker is illustrated in Fig. 2. Compared with holistic appearance model, local appearance models facilitate the tracker to maintain its stability against drastic changes of the appearances. In the presented method, the part-based strategy is incorporated into the background-aware correlation filters with different cropping matrices. After the tracking results are obtained from 5 different parts, a coarse-to-fine strategy is applied to the tracking framework. A structure comparison of parts from consecutive frames locates the tracking object coarsely. Then, a Bayesian inference framework is used to further refine the tracking results. With a novel threshold generation method depending on Gaussian process regression, each local appearance model is updated adaptively.

Remark 1: It is noted that the proposed tracking method utilized different cropping matrices to construct 5 tracked parts. The detection process is accomplished by a structure comparison and particle filter. With Gaussian process regression, a probabilistic relationship between parameters PSR and SCCM is obtained to update local appearance models.

B. BACKGROUND-AWARE CORRELATION FILTER

1) FILTER TRAINING

In the classical CF framework, the filter $\mathbf{w}$ is trained with the ridge regression approach. Its raw vectorized samples with length N are derived from image patch $\mathbf{x}$. To avoid boundary effects and gain real samples from the background, a successive cropping operation with two types of binary matrices, i.e., $\mathbf{B}$ and $\mathbf{P}$, is employed to get training examples

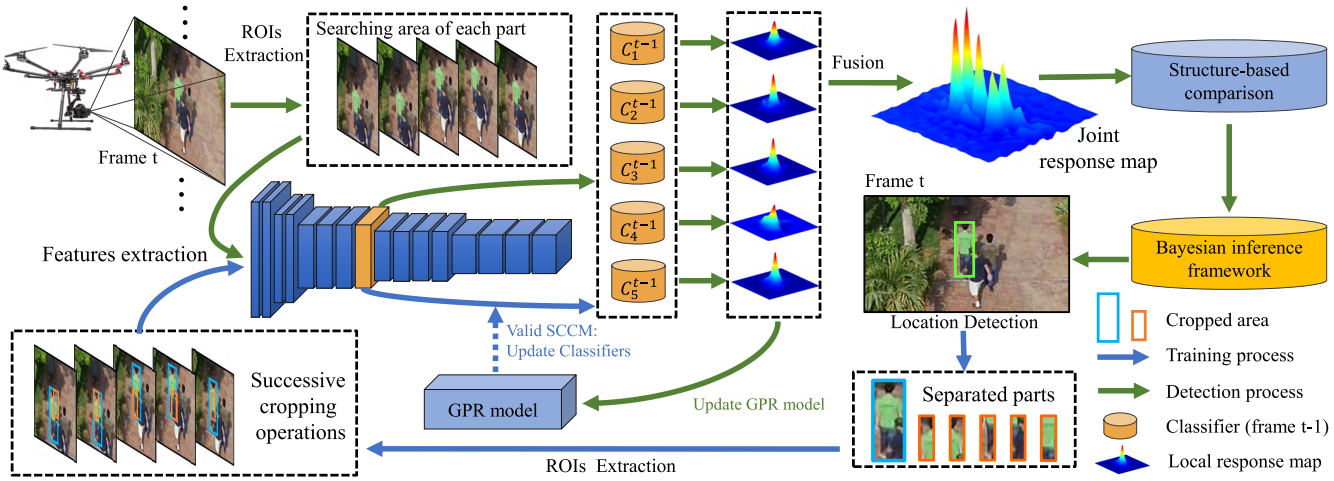


FIGURE 2. Part-based background-aware visual tracker for UAV with convolutional features. The classifier of the tracked parts, i.e., local region appearance model, is updated with an online background-aware correlation filter learned from the frame $t-1$, and then applied to estimate the local region response on the frame t.

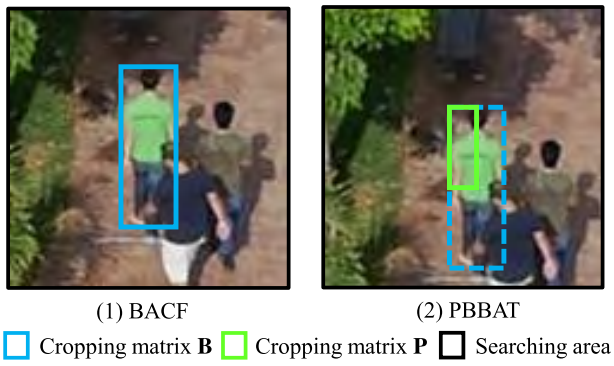


FIGURE 3. Comparison of cropping operations of each tracked part between the BACF tracker and the proposed PBBAT method.

with a smaller size M_p. The size of samples from intermediate results, i.e. from the matrix $\mathbf{B}$, is M_b, $M_p < M_b \ll N$.

Remark 2: The relative spatial location of samples cropped from the searching area is illustrated with Fig. 3. In the BACF tracker, only the whole target is cropped from the searching area. In the proposed tracker, with different $\mathbf{B}$ and $\mathbf{P}$ matrices, the tracked parts are extracted from the object after two successive cropping operations.

In the Fourier domain, the objective function $\mathcal{E}_{(i)}(\mathbf{w}_{(i)}, \hat{\mathbf{g}}_{(i)})$ of tracked part i is formulated as following to obtain desired parameters of filter $\mathbf{w}_{(i)}$:

$$\mathcal{E}_{(i)}(\mathbf{w}_{(i)}, \hat{\mathbf{g}}_{(i)}) = \frac{1}{2}\| \sum_{d=1}^{D} \hat{\mathbf{X}}_{(i)d}\hat{\mathbf{g}}_{(i)d} - \hat{\mathbf{y}}\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{D}\|\mathbf{w}_{(i)d}\|_2^2, \tag{1}$$

where $\hat{\mathbf{y}}$ is the vectorized Gaussian regression label, which is identical to each part. The subscript d denotes the d-th one of totally D feature channels. $\hat{\mathbf{X}}_{(i)}$ represents cyclic samples of the tracked part i in the Fourier domain. In detail, $\hat{\mathbf{X}}_{(i)d}$ is defined as $\hat{\mathbf{X}}_{(i)d} = \mathrm{diag}(\hat{\mathbf{x}}_{(i)d})$, which means the features of samples in the d-th feature channel in the Fourier domain. $\hat{\mathbf{g}}_{(i)}$ is an auxiliary variable, which can be formulated as $\hat{\mathbf{g}}_{(i)d} = \mathcal{F}(\mathbf{B}_{(i)}^{\top}\mathbf{P}_{(i)}^{\top}\mathbf{w}_{(i)d})$. Both the symbol $\hat{}$ and $\mathcal{F}$ represent the discrete Fourier transform. An alternative formulation is $\hat{\mathbf{g}}_{(i)d} = \sqrt{N}\mathbf{F}\mathbf{B}_{(i)}^{\top}\mathbf{P}_{(i)}^{\top}\mathbf{w}_{(i)d}$, where $\mathbf{F}$ is an orthonormal $N \times N$ mapping matrix for the Fourier transform. The $M_b \times N$ binary matrix $\mathbf{B}$ and $M_p \times M_b$ binary matrix $\mathbf{P}$ implement cropping operations together, which are able to successively crop the M_b and M_p elements from the raw signal with size N. The $\top$ denotes the conjugate transpose of a matrix or vector. λ is the trade-off coefficient for the Tikhonov regularization term. To solve the lack of closed-form solution in Eq. 1, an augmented Lagrangian method (ALM) [39] is applied. The specific Lagrangian function of part i is able to be expressed without single channel representation:

$$\begin{aligned}\mathcal{L}_{(i)}(\mathbf{w}_{(i)}, \hat{\mathbf{g}}_{(i)}, \hat{\boldsymbol{\zeta}}_{(i)}) &= \frac{1}{2}\|\hat{\mathbf{X}}_{(i)}\hat{\mathbf{g}}_{(i)} - \hat{\mathbf{y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}_{(i)}\|_2^2 \\ &+ \hat{\boldsymbol{\zeta}}_{(i)}^{\top}\left(\hat{\mathbf{g}}_{(i)} - \sqrt{N}\left(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}_{(i)}^{\top}\mathbf{P}_{(i)}^{\top}\right)\mathbf{w}_{(i)}\right) \\ &+ \frac{\mu}{2}\|\hat{\mathbf{g}}_{(i)} - \sqrt{N}(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}_{(i)}^{\top}\mathbf{P}_{(i)}^{\top})\mathbf{w}_{(i)}\|_2^2,\end{aligned} \tag{2}$$

where μ is the penalty parameter and $\hat{\boldsymbol{\zeta}}$ is the Lagrangian parameters vector in the Fourier domain. $\mathbf{I}_K$ is $K \times K$ identity matrix. By Kronecker product $\otimes$, the reformulated term is:

$$\sum_{D}\hat{\mathbf{X}}_{(i)d}\hat{\mathbf{g}}_{(i)d} = \sqrt{N}\hat{\mathbf{X}}_{(i)}(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}_{(i)}^{\top}\mathbf{P}_{(i)}^{\top})\mathbf{w}_{(i)}. \tag{3}$$

The ALM problem in Eq. 2 can be solved iteratively by alternating direction method of multipliers (ADMM). This primal problem can be separated into two subproblems, which can obtain analytic solutions, i.e., $\mathbf{w}_{(i)}^*$ and $\hat{\mathbf{g}}_{(i)}^*$, respectively. The result of the subproblem $\mathbf{w}_{(i)}^*$ is formulated

as:

$$\mathbf{w}^*_{(i)} = \frac{1}{\sqrt{T}}(\mu + \frac{\lambda}{T})^{-1}(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}^\top_{(i)}\mathbf{P}^\top_{(i)})(\mu\hat{\mathbf{g}}_{(i)} + \hat{\boldsymbol{\zeta}}_{(i)}). \quad (4)$$

Moreover, with sparse banded property and the Sherman-Morrison formula [40], ADMM iterations can make a real-time tracking performance. The solution of subproblem $\hat{\mathbf{g}}^*_{(i)}$ can be formulated as:

$$\begin{aligned}\hat{\mathbf{g}}_{(i)}(m_p)^* = \frac{1}{\mu}\big(&M_p\hat{\mathbf{y}}(m_p)\hat{\mathbf{x}}_{(i)}(m_p) \\ &- \hat{\boldsymbol{\zeta}}_{(i)}(m_p) + \mu\hat{\mathbf{w}}_{(i)}(m_p)\big) \\ &- \frac{\hat{\mathbf{x}}_{(i)}(m_p)}{\mu(\hat{\mathbf{s}}_{\mathbf{x}(i)}(m_p) + M_p\mu)}\big(M_p\hat{\mathbf{y}}(m_p)\hat{\mathbf{s}}_{\mathbf{x}(i)}(m_p) \\ &- \hat{\mathbf{s}}_{\boldsymbol{\zeta}(i)}(m_p) + \mu\hat{\mathbf{s}}_{\mathbf{w}(i)}(m_p)\big),\end{aligned} \quad (5)$$

where $m_p = [1 \cdots M_p]$ is the element number in the $\hat{\mathbf{g}}_{(i)}$. The parameters in Eq. 5 are defined as:

$$\begin{aligned}\hat{\mathbf{s}}_{\mathbf{x}(i)}(m_p) &= \hat{\mathbf{x}}_{(i)}(m_p)\hat{\mathbf{x}}_{(i)} \\ \hat{\mathbf{s}}_{\boldsymbol{\zeta}(i)}(m_p) &= \hat{\mathbf{x}}_{(i)}(m_p)\hat{\boldsymbol{\zeta}}_{(i)}. \\ \hat{\mathbf{s}}_{\mathbf{w}(i)}(m_p) &= \hat{\mathbf{x}}_{(i)}(m_p)\hat{\mathbf{w}}_{(i)}\end{aligned} \quad (6)$$

Remark 3: The formulation deviation of Eq. 5 can be found at the appendix .

Finally, the update mechanism of the $i+1$ ADMM iteration is defined as:

$$\hat{\boldsymbol{\zeta}}^{(u+1)}_{(i)} = \hat{\boldsymbol{\zeta}}^{(u)}_{(i)} + \mu(\hat{\mathbf{g}}^{(u)}_{(i)} - \hat{\mathbf{w}}^{(u)}_{(i)}). \quad (7)$$

2) OBJECT DETECTION

The location and scale changes of the tracked part in frame t is estimated with a new image patch $\mathbf{z}^t_{(i)}$ and the auxiliary variable $\hat{\mathbf{g}}^{t-1}_{(i)}$. With multiple resolutions of the searching area, a maximum correlation filter response can be determined in order to estimate the part location and scale changes:

$$\hat{\mathbf{s}}^t_{(i)} = \arg\max_{\mathbf{s}_{(i)}}\{\hat{\mathbf{z}}^t_{(i)}(\mathbf{s}_{(i)}) \odot \hat{\mathbf{g}}^{t-1}_{(i)}\}, \quad (8)$$

where $\hat{\mathbf{s}}^t$ is the expected location and scale changes of tracking object. $\odot$ denotes element-wise product.

3) FILTER UPDATING

The filter is updated with below formulation:

$$\tilde{\mathbf{x}}^t_{(i)} = (1-\alpha)\tilde{\mathbf{x}}^{t-1}_{(i)} + \alpha\mathbf{x}^t_{(i)}, \quad (9)$$

where $\tilde{\mathbf{x}}^t_{(i)}$ is appearance model that is obtained from $\mathbf{x}^t_{(i)}$ and $\tilde{\mathbf{x}}^{t-1}_{(i)}$. α is a constant learning rate. However, not frame by frame, the proposed tracking method presents a adaptive threshold, which controls the updating frequency. Gaussian process regression is applied. Details can be found in section III-F.

Remark 4: As shown in Fig. 4, the state transition graph in line (1) represents the update mechanism of the BACF tracker. it is updated frame-by-frame. The state of classifier

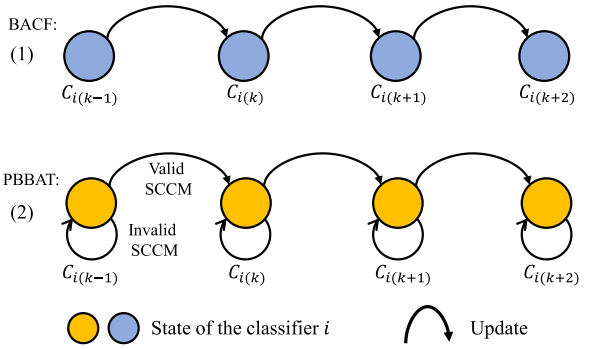


FIGURE 4. Update mechanism of BACF tracker and the proposed PBBAT tracker.

changes at every frame. It is not able to deal with object occlusion effectively. Compared with that, the proposed tracker can update the classifier adaptively, as illustrated in line (2). The tracker can maintain the stability against occlusion situations without the contamination of local appearance models.

C. RESPONSE MAP FUSION WITH ADAPTIVE WEIGHTS

As shown in Fig. 2, the tracking object is divided into multiple parts. For each part, an independent classifier, as the filters described in section III-B, is used to provide local response map $f^t_{(i)}$. Finally, these local response maps are fused into a joint response map f^t to locate the tracking object. To improve the robustness of UAV tracking, adaptive weight, i.e., the importance of each local response map, is designed based on two parameters [21]: (1) peak-to-sidelobe ratio (PSR): it evaluates the sharpness of response map. (2) smooth constraint of confidence maps (SCCM): it evaluates the smoothness of response map. The adaptive weight $\beta^t_{(i)}$ of each part is defined as:

$$\beta^t_{(i)} = \gamma\frac{1}{SCCM^t_{(i)}} + PSR^t_{(i)}, \quad (10)$$

where γ is a trade-off parameter between the sharpness and temporal consistency of the response map. Its value is defined specifically in the Table 1. $SCCM^t_{(i)}$ is the smoothness of the i-th response map on the t-th image frame. $PSR^t_{(i)}$ is the sharpness of the i-th response map on the t-th image frame. The joint response map f^t, which combines different local response maps with corresponding adaptive weights, is defined as:

$$f^t = \sum \beta^t_{(i)}f^t_{(i)}. \quad (11)$$

Remark 5: As the joint response map shown in Fig. 5, the disturbing effects of the occluded parts can be suppressed to reduce their effects for locating UAV object. The values β^t of each tracked part i are from the 49-th frame of the challenging image sequence group3-2.

D. STRUCTURE COMPARISON

A novel approach with a coarse-to-fine strategy is proposed to estimate the location and scale changes of tracking object.

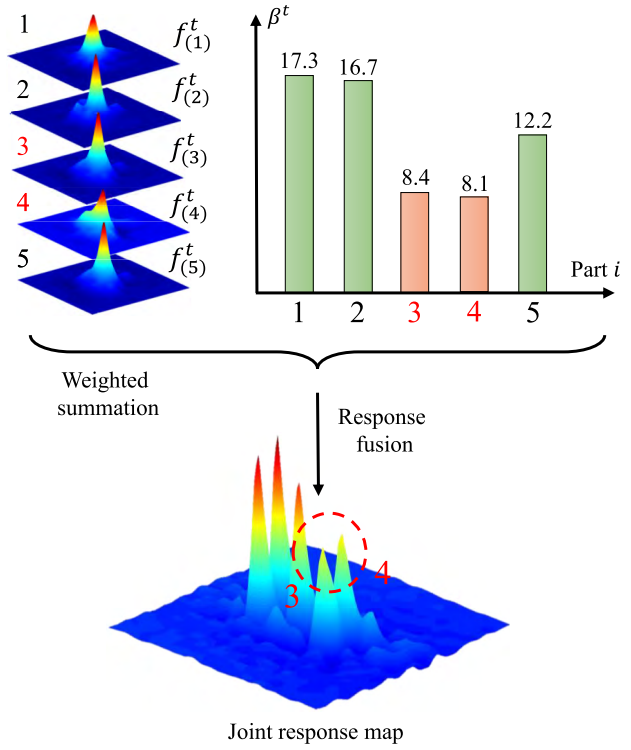


FIGURE 5. Weighted fusion of local response maps and the joint response map.

Specifically, the structures of tracking object on two consecutive frames are compared firstly for estimating the initial location and scale changes, and then a Bayesian inference framework is applied to obtain the final object location and scale changes.

After obtaining the joint response map, the coarse location and scale changes of tracking object are estimated based on the results from local response maps. To achieve a coarse estimation, the shift vectors of all parts are used to infer the result of object translation. In details, the translation is calculated with the shift vectors $\mathbf{v}^t_{(i)}$ and their trust scores $\omega^t_{(i)}$. The shift vector of the tracking object $\mathbf{v}^t$ and trust scores are defined as:

$$\mathbf{v}^t = \sum_i \omega^t_{(i)} \mathbf{v}^t_{(i)}, \tag{12}$$

$$\omega^t_{(i)} = \frac{\beta^t_{(i)}}{\sum \beta^t_{(j)}}. \tag{13}$$

In Eq. 12, a high $\omega^t_{(i)}$ represents a high trust-level of this part. The translation of the tracking object is determined with shift vectors of reliable parts. The effects of occluded parts are reduced to maintain the tracking robustness.

To estimate the scale variation, a method is proposed based on the structure of all local response maps. In this approach, the scale changes of the tracking object can refer to the distribution of its reliable local response maps. The error of

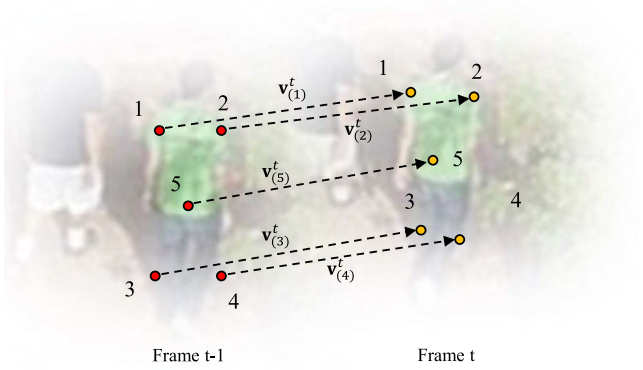


FIGURE 6. The mechanism of structure comparison. In two consecutive frames t and $t-1$, the red and orange points represent the locations of the tracking parts, i.e., the center of bounding box. The targets from two frames are drawn with its location according to each frame, respectively.

the shift vector of each part $e_{(i)}$ is defined as:

$$e_{(i)} = \|\mathbf{v}^t_{(i)} - \mathbf{v}^t\|. \tag{14}$$

The standard deviation of these errors σ_e is defined as:

$$\sigma_e = \sqrt{\frac{\sum_i (e_{(i)} - \bar{e})^2}{n_p}}, \tag{15}$$

where $\bar{e}$ is the mean value of these errors. The value n_p is the number of the part. This standard deviation represents how spread out the vectors $\mathbf{v}^t_{(i)}$ are, is calculated as the threshold to select reliable local response maps. If $e_{(i)} > \sigma_e$, the corresponding local response map will be judged as not reliable and will be discarded. The coarse scale estimation can be formulated as:

$$\triangle sc = \frac{\sigma^t_s}{\sigma^{t-1}_s}, \tag{16}$$

where σ^t_s and σ^{t-1}_s denote the standard deviation of peak locations of reliable local response maps at frame t and $t-1$. In this work, the location and scale changes are updated initially with the shift vector $\mathbf{v}^t$ and the ratio of scale variation $\triangle sc$.

Remark 6: In the UAV tracking applications, as illustrated in Fig. 6, object tracking with multiple parts mainly has three characteristics: (1) The unoccluded parts extracted from a common object mostly show a similar movement. (2) The scale of the object between two consecutive frames does not change considerably. The tracked parts can maintain a quasi-invariant structure. (3) Compared with the previous frame, most of the parts locate with a similar distribution in the current frame. Based on the characteristics above, the structure comparison can be applied to estimate the initial location and scale variation of the tracking object.

E. BAYESIAN INFERENCE FRAMEWORK

In this framework, the final object location and scale changes are estimated with the initial results obtained from structure

comparison. The object state $\mathbf{s}^t$ is formulated with affine motion, it is defined as:

$$\hat{\mathbf{s}}^t = \arg\max_{\mathbf{s}_j^t} p(\mathbf{s}_j^t|\mathbf{z}^{1:t}), \tag{17}$$

where $\mathbf{z}^{1:t}$ is the measurement set with respect to the joint confidence map, i.e., $\mathbf{z}^{1:t} = \{\mathbf{z}_i, i = 1, \cdots, k\}$. $\mathbf{s}_j^t$ is the state of the j-th sample. To model the tracking process, the Chapman-Kolmogorov equation is used, i.e.:

$$p(\mathbf{s}^t|\mathbf{z}^{1:t}) \propto p(\mathbf{z}^t|\mathbf{s}^t) \int p(\mathbf{s}^t|\mathbf{s}^{t-1}) p(\mathbf{s}^{t-1}|\mathbf{z}^{1:t-1}) d\mathbf{s}^{t-1}, \tag{18}$$

where system model $p(\mathbf{s}^t|\mathbf{s}^{t-1})$ is defined as:

$$p(\mathbf{s}^t|\mathbf{s}^{t-1}) \sim \mathcal{N}(\mathbf{s}^t, \tilde{\mathbf{s}}^{t-1}, \mathbf{\Psi}), \tag{19}$$

where $\tilde{s}^{t-1}$ is based on the coarse estimation of location and scale from the previous result, the shift vector $\mathbf{v}^t$ and the ratio $\triangle sc$. $\mathbf{\Psi}$ denotes a diagonal covariance matrix whose elements are the variances of affine parameters.

Measurement model $p(\mathbf{z}^t|\mathbf{s}^t)$ in Eq. 18 is defined as:

$$p(\mathbf{z}^t|\mathbf{s}^t) = \sum f^t(\mathbf{s}_j^t) \odot \frac{M^t}{|M^t|}, \tag{20}$$

where M^t denotes the cosine window spatial mask whose peak depends on the maximum of local response maps. $|\cdot|$ is the number of the pixels in the corresponding bounding box. $f^t(\mathbf{s}_j^t)$ is the response patch of the state $\mathbf{s}_j^t$ from joint response map.

Remark 7: The calculation of the maximum posterior $p(\mathbf{s}^t|\mathbf{z}^{1:t})$ in Eq. 17 is equivalent to obtaining the expectation of the probabilistic distribution $p(\mathbf{z}^t|\mathbf{s}^t)$, i.e. the likelihood. Traditionally, this likelihood is calculated by a set of eigen-basis vectors or methods using templates. In the proposed method, inspired by [21], the joint response map is applied, which represents the likelihood. This method significantly simplifies the computation. In Fig. 7, the response scores from each element in a bounding box with respect to each sampling candidate can be calculated efficiently as the likelihood.

F. UPDATING WITH GAUSSIAN PROCESS REGRESSION

A novel adaptive threshold is proposed to update each local appearance model, i.e., classifier. In this work, we model the relationship between *PSR* and *SCCM* with Gaussian process regression (GPR) to achieve adaptive updating. This relationship is formulated as a set of functions, i.e., $g : a \in \mathbb{R} \rightarrow g(u) \in \mathbb{R}$, where $u = PSR_{(i)}^t$ and $g(u) = SCCM_{(i)}^t$. The Gaussian process (GP) model describes the distribution of this function set:

$$g(u) \sim \mathcal{GP}\big(\mathbf{m}(u), \mathbf{G}(u, u')\big), \tag{21}$$

where $\mathcal{GP}$ denotes Gaussian process. $\mathbf{m}(u)$ and $\mathbf{G}(u, u')$ are the mean function and covariance function of this set of functions. This covariance function specifies the covariance between pairs of $PSR_{(i)}$:

$$k(q, q') = {\sigma_f}^2 \exp\big(-\frac{1}{2l^2}(q - q')^2\big), \tag{22}$$

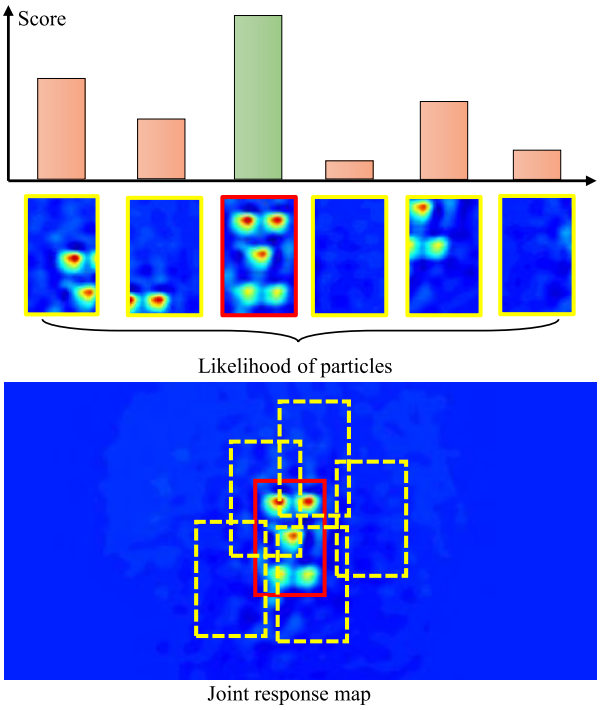


FIGURE 7. Calculation of likelihood on joint response map.

where σ_f and l are hyperparameters. q and q' are the inputs, i.e., $PSR_{(i)}$ values. After the normalization of raw inputs $PSR_{(i)}$ and outputs $SCCM_{(i)}$, the zero-mean distribution of the functions is formulated with the following prediction for each tracking part i at frame t:

$$\begin{bmatrix} \mathbf{y} \\ g_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}(\mathbf{a}, \mathbf{a}) + {\sigma_n}^2\mathbf{I} & \mathbf{k}(\mathbf{x}, a_*) \\ \mathbf{k}(a_*, \mathbf{a}) & k(a_*, a_*) \end{bmatrix}\right), \tag{23}$$

where $y = g(a) + \epsilon$ is the element of $\mathbf{y}$, which are the noisy observations of all functions. ϵ is Gaussian noise with variance matrix ${\sigma_n}^2\mathbf{I}$. a_* is normalized value of PSR_i^t and the elements of vector $\mathbf{a}$ are normalized values of previous $PSR_{(i)}$. g_* is normalized value of $SCCM_{(i)}^t$. To improve the update performance, we only select the $\mathbf{a}$ and $\mathbf{y}$ from $t - t_r$ to $t - 1$ frames, where t_r is the length of inputs memory. This approach makes the GP model focus more on the recent inputs and discard the distant ones. $\mathbf{K}(\cdot, \cdot)$ and $\mathbf{k}(\cdot, \cdot)$ denote the covariance matrix and vector of inputs, respectively. Deriving the conditional distribution $g_*|\mathbf{a}, a_*, \mathbf{y}$, the key predictive equations, which describe the distribution of the functions, are defined as:

$$\bar{g}_* = \mathbf{k}(a_*, \mathbf{a})\big[\mathbf{K}(\mathbf{a}, \mathbf{a}) + {\sigma_n}^2\mathbf{I}\big]^{-1}\mathbf{y}, \tag{24}$$

$$\mathbb{V}(g_*) = \mathbf{K}(\mathbf{a}, \mathbf{a}) - \mathbf{k}(a_*, \mathbf{a})\big[\mathbf{K}(\mathbf{a}, \mathbf{a}) + {\sigma_n}^2\mathbf{I}\big]^{-1}\mathbf{k}(\mathbf{a}, a_*), \tag{25}$$

where $\bar{g}_*$ and $\mathbb{V}(g_*)$ are the mean and variance of the conditional distribution, respectively. Taking advantage of these parameters, a valid region of *SCCM* is constructed. The upper limit of this region is $\bar{g}_* + 2\sqrt{\mathbb{V}(g_*)}$ and its lower limit is zero.

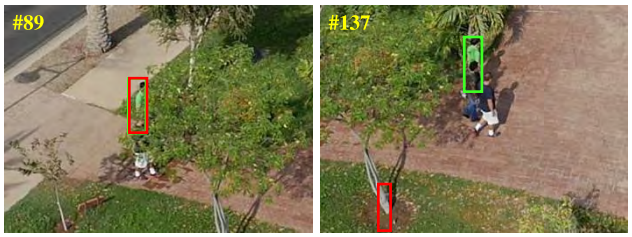


FIGURE 8. Tracking comparison with GPR and constant threshold.

On this basis, the update scheme of appearance model i at the frame t is defined as:

$$\tilde{\mathbf{x}}_{(i)}^{t} = \begin{cases} (1-\alpha)\tilde{\mathbf{x}}_{(i)}^{t-1} + \alpha \mathbf{x}_{(i)}^{t}, & \text{if } SCCM_{(i)}^{t} \text{ is valid} \\ \tilde{\mathbf{x}}_{(i)}^{t-1}, & \text{else,} \end{cases} \tag{26}$$

where α is the learning rate that controls the update velocity of the appearance model. When the calculated $SCCM_{(i)}^{t}$ is located in the valid region, $\tilde{\mathbf{x}}_{(i)}^{t}$ will be updated. Otherwise, it will not be changed.

Remark 8: In [21], the tracker is updated with a constant threshold. Compared with that, the presented adaptive threshold is able to assist the tracker to achieve a better performance, especially against partial or full occlusion, as shown in Fig. 8. The workflow of PBBAT tracker can be seen in Algorithm. 1.

Algorithm 1 PBBAT Tracker

Input: Object state $\hat{\mathbf{s}}^{t-1}$ on frame $t-1$,
Learned local appearance models $\mathbf{x}^{t-1}$
Trained correlation filers $\hat{\mathbf{g}}^{t-1}$
Output: Estimated object state $\hat{\mathbf{s}}^{t}$ on frame t

for $t = 2$ *to end* **do**
  for $i = 1$ *to* n_p **do**
    Extract the image patch $\hat{\mathbf{z}}_{(i)}^{t}$ in frame t centered at the location of part i on frame $t-1$
    Convolute the filter $\hat{\mathbf{g}}_{(i)}^{t-1}$ with $\hat{\mathbf{z}}_{(i)}^{t}$ with different scales to generate local response map $f_{(i)}^{t}$ and detect position
    Calculate $\beta_{(i)}^{t}$ with Eq.10
  end
  Fuse the local response maps with Eq. 11
  Compare the structure of locations of parts with Eq. 12, Eq. 14, Eq. 15 and Eq. 16
  Detect object state $\hat{\mathbf{s}}^{t}$ on the joint response map f^{t} using Bayesian inference framework with Eq. 19 and Eq. 20
  for $i = 1$ *to* n_p **do**
    Extract the convolutional features at the object state $\hat{\mathbf{s}}^{t}$
    Update appearance model $\mathbf{x}_{(i)}^{t}$ adaptively with the Eq. 26
    Train new filter $\hat{\mathbf{g}}_{(i)}^{t-1}$ with the Eq. 4, Eq. 5 and Eq. 7
  end
end

IV. EVALUATION OF TRACKING PERFORMANCE

In this section, the proposed PBBAT tracker is evaluated and compared with other 20 popular state-of-the-art trackers, and the limitations of PBBAT tracker are also discussed.

A. EVALUATION CRITERION

For the evaluation criterion of the tracking performance, one-pass evaluation (OPE) [16] with the precision rate (PR) and success rate (SR) are employed. For the PR, it is derived from the center location error (CLE). It is measured with the Euclidean distance in pixels between the center of the estimated object bounding box and the ground truth bounding box. Its formulation is:

$$\text{CLE} = ||C_E - C_{GT}||, \tag{27}$$

where C_E and C_{GT} represent the center of estimated object bounding box and the ground truth bounding box, respectively. In the evaluation, the successful frame is calculated with a changed threshold from 1 to 50 pixels. Then, the PR is the ratio between the number of the corresponding successful frames according to a specific CLE threshold and the number of total image frames. The precision plot (PP) is drawn with the values of PR in terms of different thresholds.

Remark 9: It is noted that a threshold $\rho = 20$ pixels is normally applied to evaluate the overall performance of a tracker.

For the SR, it depends on success score (SS), which is defined as:

$$\text{SS} = \frac{|ROI_E \cap ROI_{GT}|}{|ROI_E \cup ROI_{GT}|}, \tag{28}$$

where $|*|$ is the number of pixels in a region. $\cup$ and $\cap$ are the union and intersection operators. ROI_{GT} and ROI_E are the ground-truth and estimated regions of the tracking object. Similarly, the SR is defined as the ratio between the successful frame number with respect to a threshold and total image frame number.

Remark 10: In general, the area under the curve (AUC) of a success plot (SP) is used to evaluate the overall tracking performance.

B. OVERALL EVALUATION

In this work, the PBBAT tracker with HOG or convolutional features is evaluated and compared with other proposed 20 state-of-the-art trackers, i.e., BACF [18], HCF [28], SRDCF [20], PTAV [34], MEEM [41], MUSTer [42], Struck [14], SAMF [43], Staple [44], KCC [45], Staple-CA [46], DSST [38], CSK [47], KCF [17], DCF [17], IVT [9], TLD [48], ASLA [11], WMIL [49] and MIL [50]. To thoroughly evaluate the performances of these trackers,

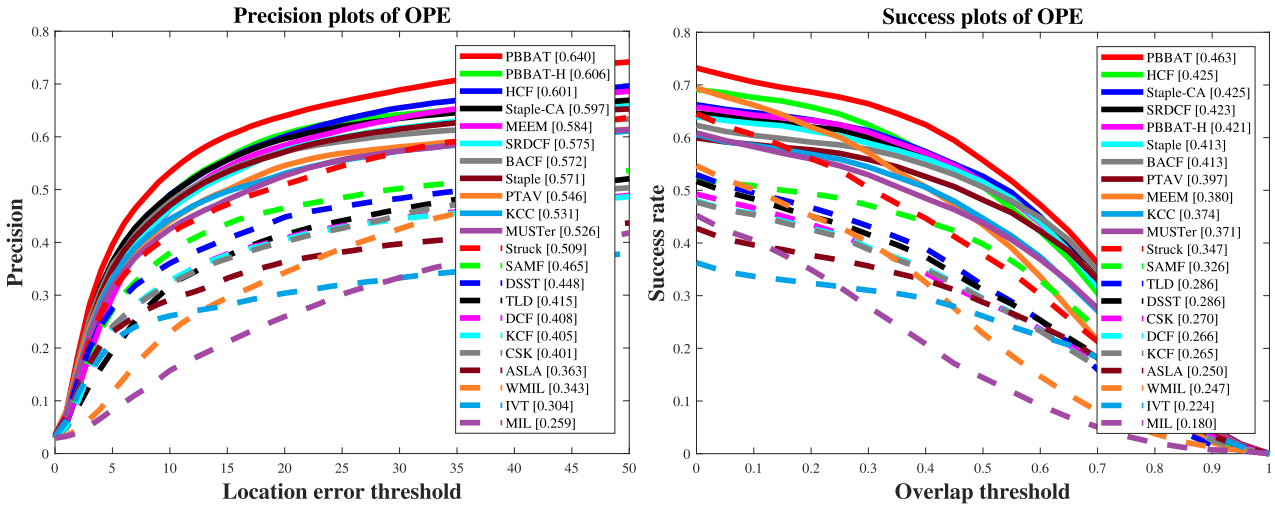


FIGURE 9. Precision and success plots of the proposed PBBAT tracker and other 20 trackers corresponding to OPE on 123 challenging UAV image sequences.

TABLE 1. Main parameters used for the proposed PBBAT tracker.

Parameters	Values
Number of parts n_p	5
trade-off coefficient γ	10^{-4}
Number of particles	300
Diagonal of covariance matrix $[\sigma_x, \sigma_y, \sigma_{sr}, \sigma_{sc}, \sigma_\theta, \sigma_\phi]$	[4,4,0,0.04,0,0]
Length of the kernel l	0.5
deviation of the signal σ_f	0.05
deviation of the noise σ_n	0.001
the used layer of VGG-19 Network	conv3-4

123 challenging aerial image sequences from [15] are employed.

Remark 11: It is noted that other 20 state-of-the-art trackers are publicly available codes or binary programs, and their default parameters provided by the authors are employed in this evaluation. For the proposed PBBAT tracker, it is implemented in MATLAB without any optimizations, its main parameters of the background-aware tracker of each part are set according to BACF tracker [18]. The parameters of particle filter and Gaussian process regression are listed in Table 1. All trackers as mentioned above are evaluated on the same computer with Intel i7-8700K CPU (3.70 GHz), 48GB RAM and NVIDIA Quadro P2000 GPU. Also, this work strictly complies with the tracker evaluation protocol from the UAV123 and calculates the average performances using PPs and SPs as the final results to conduct a fair comparison.

1) EVALUATION WITH DIFFERENT FEATURES

In this section, the PBBAT trackers with different features are compared with the BACF tracker. The overall precision plots and success plots are illustrated in Fig. 9. As can be seen in the precision plots, the scores, with a threshold $\rho = 20$ pixels, are 0.640, 0.606 and 0.572 for the PBBAT tracker with convolutional features (PBBAT), PBBAT with HOG features (PBBAT-H) and BACF tracker, respectively. Therefore, the PBBAT tracker with convolutional features performs the best. Similarly, The scores of AUC in success plots are 0.463, 0.421 and 0.413 for the PBBAT, PBBAT-H and BACF tracker. The PBBAT tracker wins the first place.

Remark 12: (1) Comparing the trackers PBBAT-H with BACF, a part-based strategy with an adaptive threshold can effectively improve the tracking performance in the UAV challenging scenarios, especially against the BACF tracker with holistic appearance model. In detail, the PBBAT-H tracker achieves a superiority of 5.9% and 1.9% according to the scores of PPs and SPs, respectively. (2) Comparing tracker PBBAT with PBBAT-H, the convolutional features provide a distinct improvement. The PBBAT tracker achieves a superiority of 5.6% and 10.0% according to the scores of PPs and SPs, respectively.

2) EVALUATION WITH OTHER STATE-OF-THE-ART TRACKERS

Fig. 9 shows the precision and success Plots of all tracking methods on 123 challenging UAV image sequences. In the precision plots, the scores of all trackers on the threshold $\rho = 20$ pixels are 0.640 (PBBAT), 0.606 (PBBAT-H), 0.601 (HCF), Stable-CA (0.597), 0.584 (MEEM), 0.575 (SRDCF), 0.572 (BACF), 0.571 (Staple), 0.546 (PTAV), 0.531 (KCC), 0.526 (MUSTer), 0.509 (Struck), 0.465 (SAMF), 0.448 (DSST), 0.415 (TLD), 0.405 (KCF), 0.401 (CSK), 0.363 (ASLA), 0.343 (WMIL), 0.304 (IVT) and 0.259 (MIL), respectively. Obviously, the PBBAT tracker has achieved the best precision against other 20 trackers. In all the success plots, the AUC-based scores of all trackers are 0.463 (PBBAT), 0.425 (HCF), 0.425 (Staple-CA), 0.423 (SRDCF), 0.421 (PBBAT-H), 0.413 (Staple), 0.413 (BACF), 0.397 (PTAV), 0.380 (MEEM), 0.374 (KCC), 0.371 (MUSTer), 0.347 (Struck), 0.326 (SAMF), 0.286 (TLD),

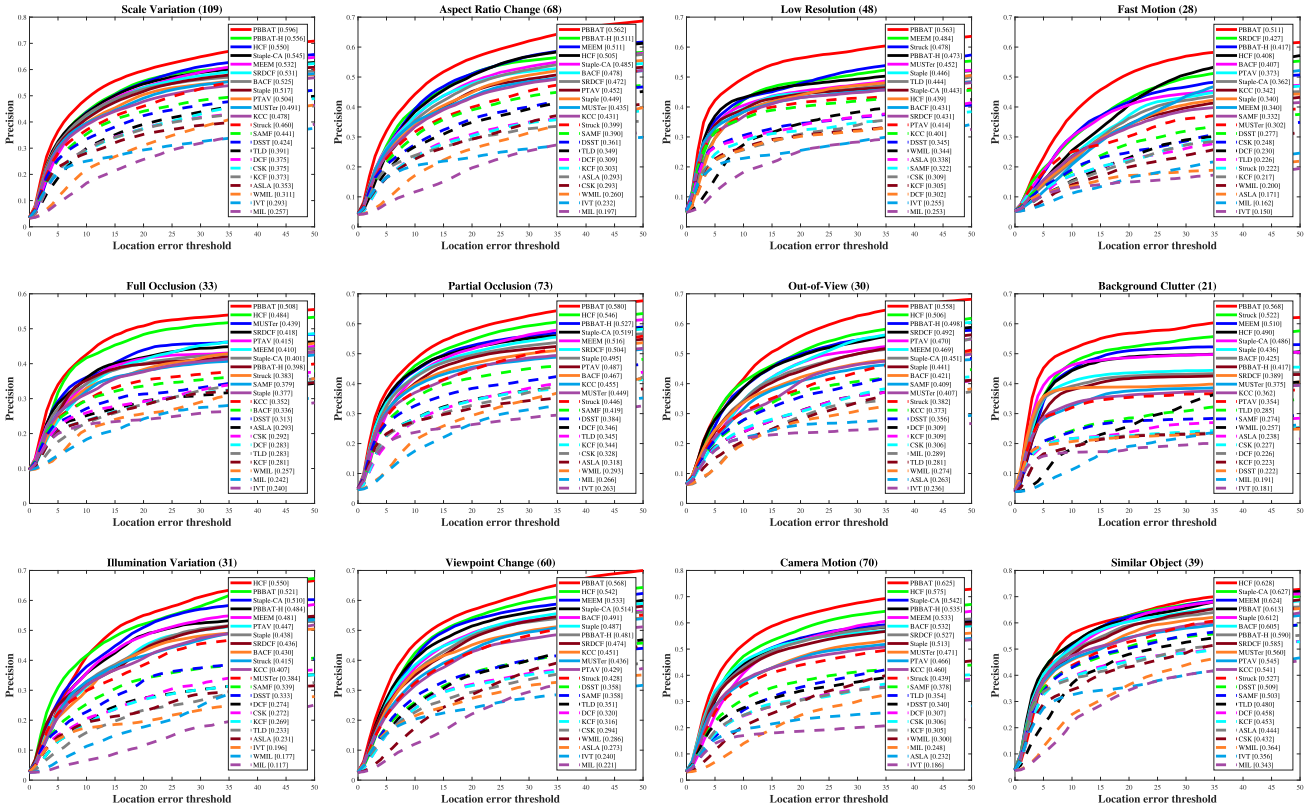


FIGURE 10. Precision plots on different attributes of the presented PBBAT tracker and other 20 state-of-the-art trackers corresponding to OPE.

TABLE 2. Scores of precision plots ($\rho = 20$ pixels) in terms of 12 challenging attributes. **Red**, blue and green fonts indicate the best, second best and third best performances amongst the PBBAT tracker and other 20 trackers.

	SV	ARC	LR	FM	FOC	POC	OV	BC	IV	VC	CM	SOB
IVT	29.3	23.2	25.5	15.0	24.0	26.3	23.6	18.1	19.6	24.0	18.6	35.6
ASLA	35.3	29.3	33.8	17.1	29.3	31.8	26.3	23.8	23.1	27.3	23.2	44.4
KCF	37.3	30.3	30.5	21.7	28.1	34.4	30.9	22.3	26.9	31.6	30.5	45.3
CSK	37.5	29.3	30.9	24.8	29.2	32.8	30.6	22.7	27.2	29.4	30.6	43.2
DCF	37.5	30.9	30.2	23.0	28.3	34.6	30.9	22.6	27.4	32.0	30.7	45.8
TLD	39.1	34.9	44.4	22.6	28.3	34.5	28.1	28.5	23.3	35.1	35.4	48.0
DSST	42.4	36.1	34.5	27.7	31.3	38.4	35.6	22.2	33.3	35.8	34.0	50.9
SAMF	44.1	39.0	32.2	33.2	37.9	41.9	40.9	27.4	33.9	35.8	37.8	50.3
Struck	46.0	39.9	47.8	22.2	38.3	44.6	38.2	52.2	41.5	42.8	43.9	52.7
KCC	47.8	43.1	40.1	34.2	35.2	45.5	37.3	36.2	40.7	45.1	46.0	54.1
MUSTer	49.1	43.5	45.2	30.2	43.9	44.9	40.7	37.5	38.4	43.6	47.1	56.0
PTAV	50.4	45.2	41.4	37.3	41.5	48.7	47.0	35.4	44.7	42.9	46.6	54.5
Staple	51.7	44.9	44.6	34.0	37.7	49.5	44.1	43.6	43.8	48.7	51.3	61.2
BACF	52.5	47.8	43.1	40.7	33.6	46.7	42.1	42.5	43.0	49.1	53.2	60.5
SRDCF	53.1	47.2	43.1	42.7	41.8	50.4	49.2	38.9	43.6	47.4	52.7	58.5
MEEM	53.2	51.1	48.4	34.0	41.0	51.6	46.9	51.0	48.1	53.3	53.3	62.4
Staple-CA	54.5	48.5	44.3	36.2	40.1	51.9	45.1	48.6	51.0	51.4	54.2	62.7
WMIL	31.1	26.0	24.4	20.0	25.7	29.3	27.4	25.7	17.7	28.6	30.0	36.4
MIL	25.7	19.7	25.3	15.0	24.0	26.3	28.9	19.1	11.7	22.1	18.6	34.3
HCF	55.0	50.5	43.9	40.8	48.4	54.6	50.6	49.0	**55.0**	54.2	57.5	**62.8**
PBBAT	**59.6**	**56.2**	**56.3**	**51.1**	**50.8**	**58.0**	**55.8**	**56.8**	52.1	**56.8**	**62.5**	61.3

0.286 (DSST), 0.270 (CSK), 0.266 (DCF), 0.265 (KCF), 0.250 (ASLA), 0.247 (WMIL), 0.224 (IVT) and 0.180 (MIL), respectively. The PBBAT tracker still ranks No. 1 against other 20 tracking methods. Thus, it can be summarized that the PBBAT tracker is better than other 20 state-of-the-art trackers according to precision and success ratio.

Remark 13: The improvements of the PBBAT tracker show that the tracking framework with a part-based strategy is able

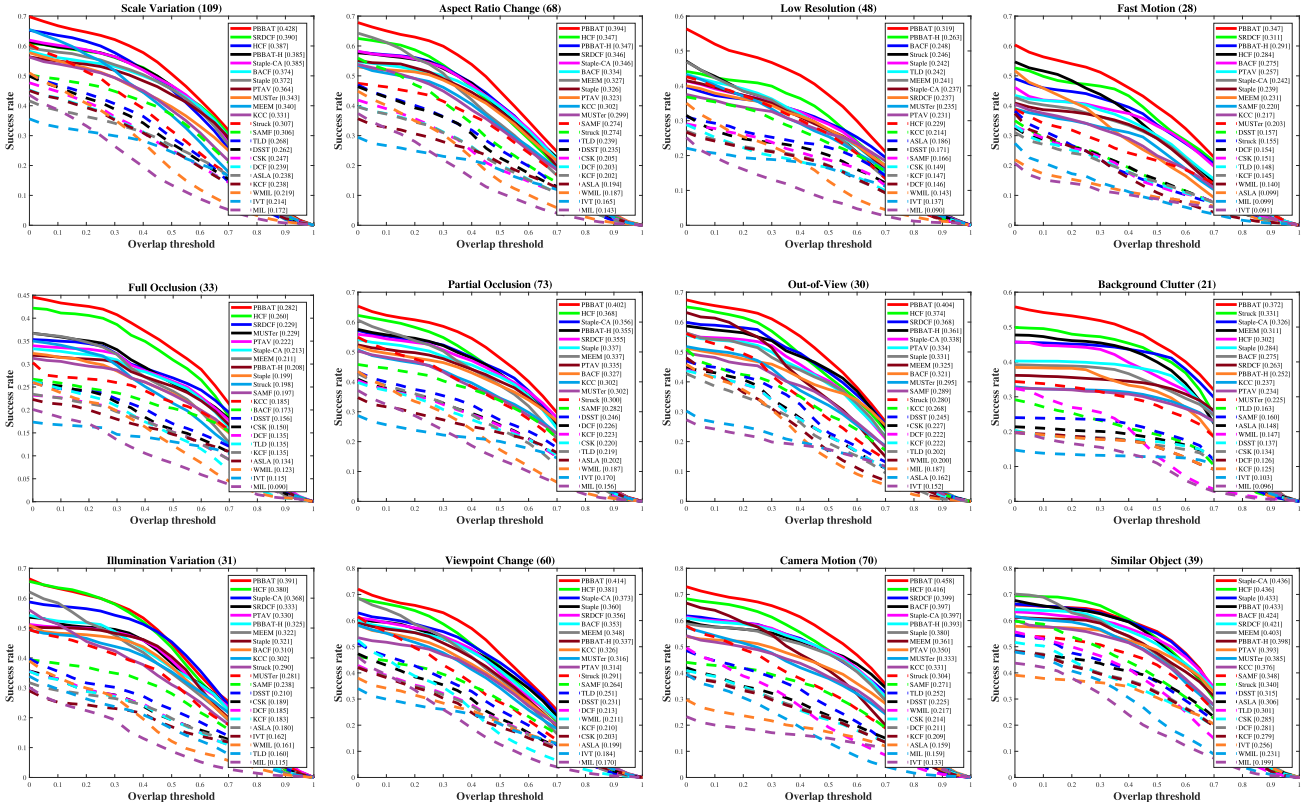


FIGURE 11. Success plots on different attributes of the proposed PBBAT tracker and other 20 state-of-the-art trackers corresponding to OPE.

TABLE 3. Scores of success plots (AUC) in terms of 12 different attributes. Red, blue and green fonts indicate the best, second best and third best performances among the PBBAT tracker and other 20 trackers.

	SV	ARC	LR	FM	FOC	POC	OV	BC	IV	VC	CM	SOB
IVT	21.4	16.5	13.7	9.1	11.5	17.0	15.2	10.3	16.2	18.4	13.3	25.6
ASLA	23.8	19.4	18.6	9.9	13.4	20.2	16.2	14.8	18.0	19.9	15.9	30.6
KCF	23.8	20.2	14.7	14.5	13.5	22.3	22.2	12.5	18.3	21.0	20.9	27.9
CSK	24.7	20.5	14.9	15.1	15.0	22.0	22.7	13.4	18.9	20.3	21.4	28.5
DCF	23.9	20.3	14.6	15.4	13.5	22.6	22.2	12.6	18.5	21.3	21.1	28.1
TLD	26.8	23.9	24.2	14.8	13.5	21.9	20.2	16.3	16.0	25.1	25.2	30.1
DSST	26.2	23.5	17.1	15.7	15.6	24.6	24.5	13.7	21.0	23.1	22.5	31.5
SAMF	30.6	27.4	16.6	22.0	19.7	28.2	28.9	16.0	23.8	26.4	27.1	34.8
Struck	30.7	27.4	24.6	15.5	19.8	30.0	28.0	33.1	29.0	29.1	30.4	34.0
KCC	33.1	30.2	21.4	21.7	18.5	30.2	26.8	23.7	30.2	32.6	33.1	37.6
MUSTer	34.3	29.9	23.5	20.3	22.9	30.2	29.5	22.5	28.1	31.6	33.3	38.5
PTAV	36.4	32.3	23.1	25.7	22.2	33.5	33.4	23.4	33.0	31.4	35.0	39.3
Staple	37.2	32.6	24.2	23.9	21.3	33.7	33.1	28.4	32.1	36.0	38.0	43.3
BACF	37.4	33.4	24.8	27.5	17.3	32.7	32.1	27.5	31.0	35.3	39.7	42.4
SRDCF	39.0	34.6	23.7	31.1	22.9	35.5	36.8	26.3	33.3	35.6	39.9	42.1
MEEM	34.0	32.7	24.1	23.1	21.1	33.7	32.5	31.1	32.2	34.8	36.1	40.3
Staple-CA	38.5	34.6	23.7	24.2	21.3	35.6	33.8	32.6	36.8	37.3	39.7	**43.6**
WMIL	21.9	18.7	14.3	14.0	12.3	18.7	20.0	14.7	16.1	21.1	21.7	23.1
MIL	17.2	14.3	9.0	9.9	9.0	15.6	18.7	9.6	11.5	17.0	15.9	19.9
HCF	38.7	34.7	22.9	28.4	26.0	36.8	37.4	30.2	38.0	38.1	41.6	43.6
PBBAT	**42.8**	**39.4**	**31.9**	**34.7**	**28.2**	**40.2**	**40.4**	**37.2**	**39.1**	**41.4**	**45.8**	43.3

to obtain a better tracking performance against other 20 state-of-the-art trackers. Some examples of tracking results are shown in Fig. 12. The code and the video of tracking results are https://github.com/vision4robotics/PBBAT-Tracker and https://youtu.be/4v5ob9YzYgE.

3) ATTRIBUTE-BASED EVALUATION AND COMPARISON

In UAV tracking applications, 12 different challenging attributes are considered, namely scale variation (SV), aspect ration change (ARC), camera motion (CM), low resolution (LR), full occlusion (FOC), fast motion (FM),

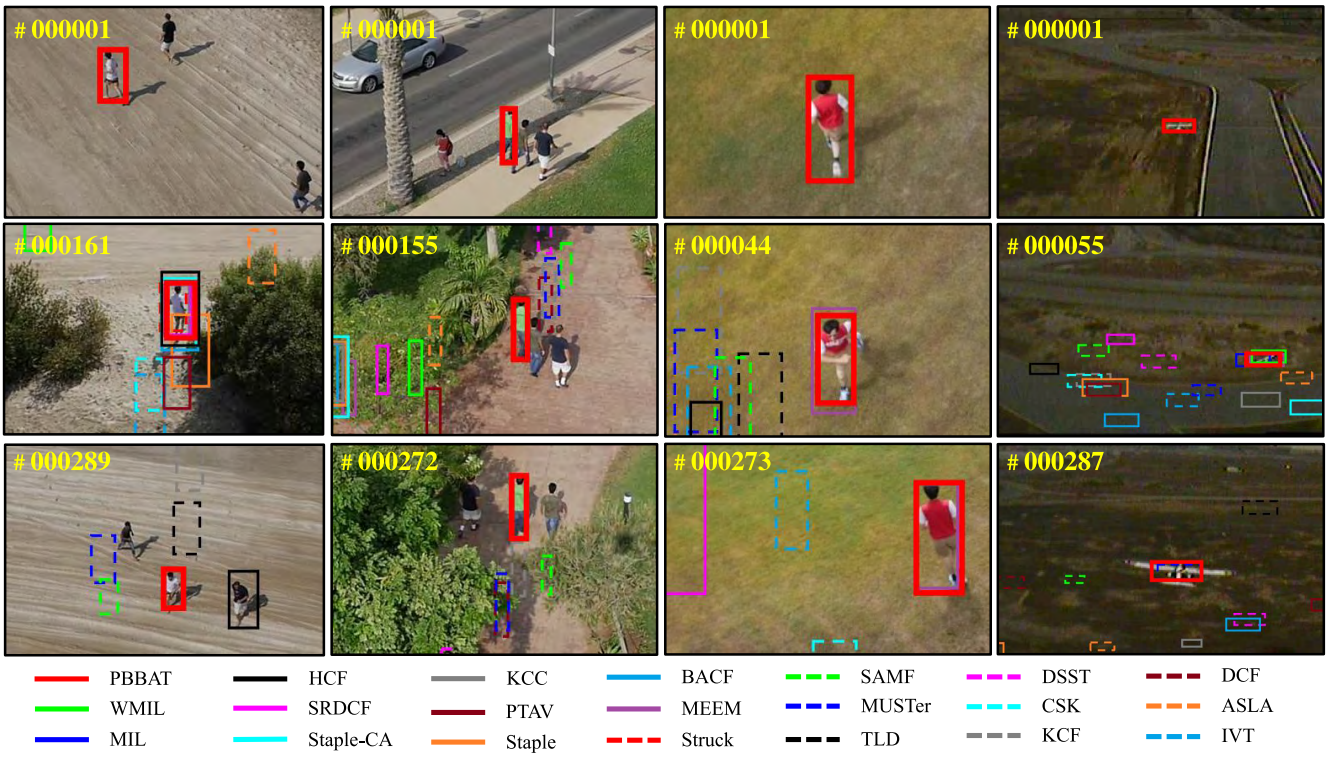


FIGURE 12. Examples of the UAV tracking results. The first, second, third, and fourth columns show the challenging image sequences from *group2-2*, *group3-2*, *perosn7-2* and *uav1-1*.

out-of-view (OV), illumination variation (IV), partial occlusion (POC), background clutter (BC), similar object (SOB), and viewpoint change (VC). For precision, as shown in Fig. 10, PBBAT outperforms all of the competing trackers on 10 attributes, i.e., ARC, VC, SV, POC, OV, FOC, FM, CM, BC, LR. Similarly, Fig. 11 shows that PBBAT has achieved the best success ratio performance in terms of the ARC, CM, OV, FOC, POC, FM, SV, IV, LR, BC and VC attributes. It is noteworthy that the proposed PBBAT tracker can crop with challenging LR, POC and FOC attributes preferably in both PPs and SPs.

Remark 14: Detailed scores in terms of 12 different attributes are provided in Table 2 and Table 3. Especially, the proposed PBBAT tracker can achieve superior performance when tracking the object with low resolution and occlusion. In detail with LR, it obtains an improvement of 36.7 % compared with the BACF tracker at the second place by SP and superiority of 16.3 % compared with the MEEM tracker by PP.

C. LIMITATIONS OF THE PROPOSED TRACKER

1) ATTRIBUTES

As illustrated in Fig. 10, the proposed PBBAT tracker does not win the first three places in the SOB attribute. Also, it only wins second place in the attribute IV. That means there is not yet a huge advantage to crop with the distraction by similar objects and illumination variation issues. Similarly, in Fig. 11, the proposed PBBAT tracker performs not the best dealing with the disturbing of similar objects.

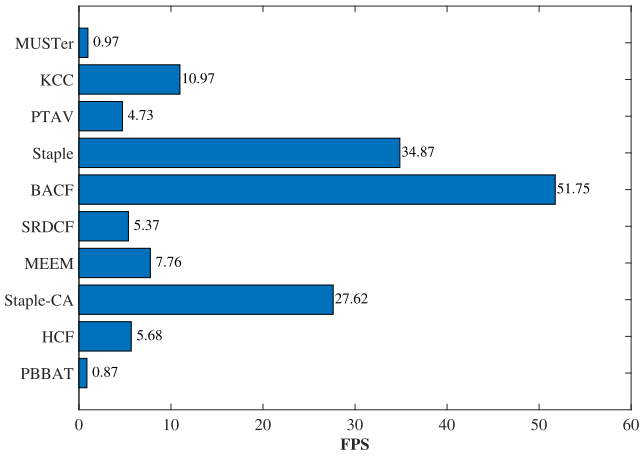


FIGURE 13. The FPS evaluation of the best 10 trackers in overrall evaluations (PPs and SPs).

2) SPEED

The proposed PBBAT tracker is implemented on the MATLAB platform with no optimization. Therefore, as shown in Fig. 13, the frame per second achieves only on average 0.82 on the 123 challenging aerial image sequences on the platform mentioned before. Fortunately, the UAV is capable of carrying high-performance GPU and CPU, for instance, DJI S1000+. With the implementation on the C platform and proper optimization for GPU and thorough usage of the parallel computation, PBBAT tracker can be applied to UAV tracking.

V. CONCLUSION

In this paper, a novel part-based background-aware visual tracker has been presented and applied for the UAV applications. Specifically, several background-aware CF trackers are used to achieve a better tracking performance compared with the classical CF tracker. An effective coarse-to-fine strategy with structure comparison and Bayesian inference framework is developed to improve the estimation of the tracking object location and scale variation. Furthermore, an adaptive threshold is established to update each local appearance model with a Gaussian process regression approach. The extensive experiments on 123 challenging UAV image sequences show that our presented visual tracker outperforms the most promising state-of-the-art visual trackers, and overcome the object appearance change caused by different challenging situations. We believe our approach will open the doors to their wider use in real-world UAV tracking tasks.

APPENDIX
THE FORMULATION DEVIATION

In this Appendix, the formulation deviation of $\hat{\mathbf{g}}^*_{(i)}$ in Eq. 5 is provided:

Subproblem $\hat{\mathbf{g}}_{(i)}$:

$$\begin{aligned}\hat{\mathbf{g}}^*_{(i)} = \arg\min_{\hat{\mathbf{g}}_{(i)}} \Big\{ & \frac{1}{2}\|\hat{\mathbf{X}}_{(i)}\hat{\mathbf{g}}_{(i)} - \hat{\mathbf{y}}\|_2^2 \\ & + \hat{\boldsymbol{\zeta}}^\top_{(i)}\Big(\hat{\mathbf{g}}_{(i)} - \sqrt{N}\big(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}^\top_{(i)}\mathbf{P}^\top_{(i)}\big)\mathbf{w}_{(i)}\Big) \\ & + \frac{\mu}{2}\|\hat{\mathbf{g}}_{(i)} - \sqrt{N}(\mathbf{I}_K \otimes \mathbf{F}\mathbf{B}^\top_{(i)}\mathbf{P}^\top_{(i)})\mathbf{w}_{(i)}\|_2^2\Big\}, \end{aligned} \quad (29)$$

which can be reformulated with each element m_p as:

$$\begin{aligned}\hat{\mathbf{g}}_{(i)}(m_p)^* = \arg\min_{\hat{\mathbf{g}}_{(i)}(m_p)} \Big\{ & \frac{1}{2}\|\hat{\mathbf{x}}^\top_{(i)}(m_p)\hat{\mathbf{g}}_{(i)}(m_p) - \hat{\mathbf{y}}(m_p)\|_2^2 \\ & + \hat{\boldsymbol{\zeta}}^\top_{(i)}\Big(\hat{\mathbf{g}}_{(i)}(m_p) - \hat{\mathbf{w}}_{(i)}(m_p)\Big) \\ & + \frac{\mu}{2}\|\hat{\mathbf{g}}_{(i)}(m_p) - \hat{\mathbf{w}}_{(i)}(m_p)\|_2^2\Big\}. \end{aligned} \quad (30)$$

The solution of Eq. 30 is obtained when its derivative is equal to zero, which is formulated as:

$$\begin{aligned}&\hat{\mathbf{x}}_{(i)}(m_p)\big(\hat{\mathbf{y}}_{(i)}(m_p) - \hat{\mathbf{x}}^\top_{(i)}(m_p)\hat{\mathbf{g}}_{(i)}(m_p)\big) \\ &\quad + T\hat{\boldsymbol{\zeta}}_{(i)}(m_p) + T\mu(\hat{\mathbf{g}}_{(i)}(m_p) - \hat{\mathbf{w}}_{(i)}(m_p)) = 0. \end{aligned} \quad (31)$$

The solution is:

$$\begin{aligned}\hat{\mathbf{g}}_{(i)}(m_p)^* = & (\hat{\mathbf{x}}_{(i)}(m_p)\hat{\mathbf{x}}^\top_{(i)}(m_p) + T\mu\mathbf{I}_K)^{-1} \\ & \times\big(\hat{\mathbf{y}}_{(i)}(m_p)\hat{\mathbf{x}}_{(i)}(m_p) - T\hat{\boldsymbol{\zeta}}_{(i)}(m_p) + T\mu\hat{\mathbf{w}}_{(i)}(m_p)\big). \end{aligned} \quad (32)$$

With the Sherman-Morrison formula, the result from Eq. 5 can be calculated.

REFERENCES

[1] M. Mueller, G. Sharma, N. Smith, and B. Ghanem, "Persistent aerial tracking system for UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1562–1569.

[2] A. Carrio, C. Fu, J.-F. Collumeau, and P. Campoy, "SIGS: Synthetic imagery generating software for the development and evaluation of vision-based sense-and-avoid systems," *J. Intell. Robotic Syst.*, vol. 84, no. 1, pp. 559–574, 2016.

[3] J. L. Sanchez-Lopez, S. Saripalli, P. Campoy, J. Pestana, and C. Fu, "Toward visual autonomous ship board landing of a VTOL UAV," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2013, pp. 779–788.

[4] X. Xue, Y. Li, and Q. Shen, "Unmanned aerial vehicle object tracking by correlation filter with adaptive appearance model," *Sensors*, vol. 18, no. 9, p. 2751, 2018.

[5] X. Li, B. Yan, H. Wang, X. Luo, Q. Yang, and W. Yan, "Corner detection based target tracking and recognition for UAV-based patrolling system," in *Proc. IEEE Int. Conf. Inf. Automat. (ICIA)*, Aug. 2016, pp. 282–286.

[6] G. Zhou, J. Yuan, I.-L. Yen, and F. Bastani, "Robust real-time UAV based power line detection and tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 744–748.

[7] 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 Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 510–521, Mar. 2016.

[8] 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 *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2014, pp. 649–655.

[9] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[10] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 657–664.

[11] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[12] C. Fu, R. Suarez-Fernandez, M. A. Olivares-Mendez, and P. Campoy, "Real-time adaptive multi-classifier multi-resolution visual tracking framework for unmanned aerial vehicles," *IFAC Proc. Volumes*, vol. 46, no. 30, pp. 99–106, 2013.

[13] C. Fu, A. Carrio, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Robust real-time vision-based aircraft tracking from unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2014, pp. 5441–5446.

[14] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 263–270.

[15] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 445–461.

[16] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[18] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1152.

[19] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4630–4638.

[20] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[21] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4902–4912.

[22] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 353–361.

[23] R. Yao, S. Xia, Z. Zhang, and Y. Zhang, "Real-time correlation filter tracking by efficient dense belief propagation with structure preserving," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 772–784, Apr. 2017.

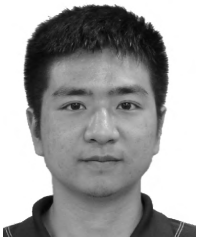
[24] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1849–1861, Jun. 2017.

[25] T. Zhang, C. Xu, and M.-H. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 365–378, Feb. 2019.
[26] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 212–219.
[27] J. Kwon and K. M. Lee, "Highly nonrigid object tracking via patch-based dynamic appearance modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2427–2441, Oct. 2013.
[28] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.
[29] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4844–4853.
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[31] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014, pp. 1–12.
[32] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.
[33] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 749–765.
[34] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5486–5494.
[35] H. Cheng, L. Lin, Z. Zheng, Y. Guan, and Z. Liu, "An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1732–1738.
[36] R. Li, M. Pang, C. Zhao, G. Zhou, and L. Fang, "Monocular long-term target following on UAVs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 29–37.
[37] Z. Qu, X. Lv, J. Liu, L. Jiang, L. Liang, and W. Xie, "Long-term reliable visual tracking with UAVs," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2000–2005.
[38] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014, pp. 4310–4318.
[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[40] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *Ann. Math. Statist.*, vol. 21, no. 1, pp. 124–127, 1950.
[41] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 188–203.
[42] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 749–758.
[43] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration" in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2015, pp. 254–265.
[44] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.
[45] C. Wang, L. Zhang, L. Xie, and J. Yuan, "Kernel cross-correlator," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 4179–4186.
[46] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1396–1404.
[47] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 702–715.
[48] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[49] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognit.*, vol. 46, pp. 397–411, Jan. 2013.
[50] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 983–990.

CHANGHONG FU received the Ph.D. degree in robotics and automation from the Computer Vision and Aerial Robotics Laboratory, Technical University of Madrid, Spain. During his Ph.D., he held two research positions at Arizona State University, Tempe, AZ, USA, and Nanyang Technological University (NTU), Singapore. After received his Ph.D. degree, he was with NTU as Postdoctoral Research Fellow. He has worked on two international, two national, and four industrial projects related to the vision for the unmanned aerial vehicle (UAV). He is currently an Assistant Professor with the School of Mechanical Engineering, Tongji University, China, and leading four projects related to the vision for multi-UAV. His research areas are intelligent vision and control for multi-UAV in complex environments.

YINQIANG ZHANG received the B.Eng. degree in mechanical engineering from Tongji University, Shanghai, China. He is currently pursuing the M.Sc. degree in mechatronics and information technology with the Technical University of Munich (TUM), Munich, Germany. His research interests include visual tracking and computer vision.

ZIYUAN HUANG is currently pursuing the B.Eng. degree in vehicle engineering, with a specialization in vehicle electronics, with Tongji University, where he is currently a Senior Student. His research interests involve visual tracking for unmanned aerial vehicles and computer vision.



RAN DUAN received the M.Sc. degree in computer vision from the European VIBOT Programme, University of Burgundy, France. He is currently pursuing the Ph.D. degree with the Adaptive Robotic Controls Lab (ArcLab), The Hong Kong Polytechnic University (PolyU), Hong Kong. During his M.Sc., he held a research position with the University of Strasbourg, France. After receiving his M.Sc. degree, he was with the Nanyang Technological University (NTU) as Research Associate, and The City College of New York as a Visiting Scholar. His research interests include computer vision, robotics, and deep learning.

ZONGWU XIE received the B.S. degree in electrical engineering and automation from the Harbin University of Science and Technology, Harbin, China, in 1996, and the M.S. and Ph.D. degrees in mechanical engineering from the Harbin Institute of Technology, Harbin, in 2000 and 2003, respectively. He is currently a Research Fellow in mechanical engineering with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His current research interest includes the design and control of robotic systems.