

# Real-time UAV tracking based on PSR stability

Yong Wang

University of Ottawa

75 Laurier Ave E, Ottawa, ON K1N 6N5

ywang6@uottawa.ca

Lu Ding

School of Aeronautics and Astronautics, Shanghai Jiao Tong University  
Shanghai, China

dinglu@sjtu.edu.cn

Robert Laganière

University of Ottawa

75 Laurier Ave E, Ottawa, ON K1N 6N5

laganier@eecs.uottawa.ca

## Abstract

*In this paper, a real-time unmanned aerial vehicle (UAV) tracking method is proposed. This approach builds a target representation from histograms of oriented gradient (HOG) and ColorNames features. Correlation filters have been utilized in tracking recently because of their high efficiency. To better fuse the tracking results from different features, peak-to-sidelobe ratio (PSR) is employed to evaluate robustness of our trackers. A stability measure is proposed, based on the PSR values computed over a short period of time which is also used to predict object position. Additionally, we show that the proposed PSR stability enables our tracking method to be robust to various appearance variations. The method is carried out on five UAV tracking datasets and achieves appealing results comparable to state-of-the-art trackers but at a lower computational complexity.*

## 1. Introduction

Unmanned aerial vehicles (UAV) equipped with cameras produce a vast amount of visual information that can be exploited in many applications [41, 45], e.g., action recognition, pedestrian detection. UAV based tracking plays an important role in these applications. For object-following applications, a UAV must track a target as it moves through an environment. For autonomous flying, a UAV must track dynamic obstacles to predict where they are moving and es-

timate their future positions.

UAV based tracking presents its own properties. The videos are captured from a high perspective in outdoor settings. The target appearance changes severely. In addition, targets are often captured at low resolution and present important illumination variations which makes UAV tracking more challenging than generic tracking.

Feature descriptor is a key component in visual tracking [1]. In order to represent an object accurately, various features have been developed, e.g., histogram of oriented gradient (HOG) [37] and color descriptors [31]. Recently, deep features have achieved significant success in generic tracking [3, 4, 5] and UAV tracking [6, 7]. Deep features are trained on large scale datasets. Consequently they are able to characterize object appearance accurately. However, extracting the deep features imply high computation and memory requirements. In UAV tracking, the tracker output should be accurate and robust, while running in real-time under limited hardware capabilities. Thus, hand-crafted features (e.g., HOG and color) constitute often a more suitable choice.

To improve robustness, backward tracking is usually employed. This scheme is useful to eliminate false tracking points [8], fuse multiple independent trackers [5] and correct tracker results [9] as the inverse temporal order provides new information to the tracker. In [8, 5], only the adjacent frame information is used. In [9] a period of history is exploited, as a temporal sequence contains more information. In [4] the responses of a short period frames are treated as a time series to better estimate the fusion weight.

In [10] tracking results are improved by using the models from previous frames to update the current target model.

Many algorithms proposed to improve tracking to determine the fusion weight of the trackers. In [11] the authors use the foreground and background differences to determine the fusion weights. Response values are employed in [4] to predict fusion weights. Peak-to-sidelobe ratio (PSR) is used in [30] to fuse different parts trackers. Overlap is employed in [5] to compare the differences between forward tracking and backward tracking. These indicate that tracking in inverse order improves temporal consistency.

Correlation filters are utilized in visual tracking because of their fast computation and high accuracy [16, 17, 18]. Correlation response is an important clue for identifying new target location. Peak-versus-noise ratio based on the correlation response is adopted in [12] to adaptively update the tracking model.

Motivated by the works mentioned above, we propose a PSR based UAV tracking method. HOG [37] and ColorNames [31, 36] features are utilized to extract features. The hand-crafted features are complementary to each other and can be calculated in a short time which is an advantage in real-time tracking. These features are tracked independently in a correlation filter framework. Backward tracking is then carried out. A PSR stability measure is defined to assess the tracking quality. The fusion weights of each tracker are computed based on the PSR stability. Finally, the position is obtained by fusing the results of independent trackers.

The contributions of this paper are as follow.

First, a PSR stability criteria is proposed to estimate the performance of different feature tracking results and assign proper weights to them. Specifically, PSR values are used to measure reliabilities of the tracking results. The reliabilities can be treated as weights to fuse different feature tracking results.

Second, backward tracking is utilized over a period of time. Different from previous work [5, 8] which only use two consecutive frames, we use the backward tracking results in multiple frames.

Third, we show how the PSR stability can be used as an indicator to effectively update target model.

Fourth, our tracker is evaluated on five UAV tracking datasets and achieved comparable results to state-of-the-art methods while running at a much higher frame rate.

## 2. Related work

There is a plethora of tracking literature. For a comprehensive review, please refer to [13, 14]. In this section, we only focus on the works most related to ours.

### 2.1. Correlation filter tracking

Recently, correlation filter tracking attracts more attention because of its fast computation and accurate results. Correlation filters were first introduced to visual tracking in [16] and intensity features were used for target representation. Kernel methods are integrated into correlation filters to achieve more robust tracking [17]. The input features are extended from a single channel (e.g., intensity) to multiple channels [18]. HOG and color features are integrated into correlation filter to improve tracking results [19]. In [3] the authors use deep features from three convolutional layers contains semantic information and spatial details. Tracking task is decomposed into translation and scale estimation to deal the problem of long-term tracking [21]. In addition, a re-detection scheme based on random fern is employed to re-detect target in case of tracking lost. A re-detection module is proposed in [22] to re-evaluate results in each frame. This module refines the tracking results and help updates object model adaptively.

To handle scale variation, a method that search over the scale space is presented in fDSST [24]. A Siamese tracking is combined with fDSST to achieve collaborative tracking [23]. Region proposal which is usually used in object detection is incorporated into Siamese tracking [25].

Background information is considered to deal with the problem of boundary effects in [26]. This method is extended in [6] by incorporating hard negative examples into training. Interpolation method is utilized in [27]. A convolution operator is used to reduce the number of parameters in the tracking model. The boundary effects are alleviated in SRDCF [28] by penalizing correlation filter coefficients according to spatial location. This approach has been improved in [29] with a joint function loss to reduce the influence of contaminated training examples.

Response of correlation filter is not only used to locate target position but also used to infer reliability of tracking results. The PSR is adopted in [30] to measure the tracking quality of each part. A PNR criterion is proposed based on the correlation response in [12] for efficient updating model.

### 2.2. Tracking fusion

In [32] a factorial Hidden Markov Model (HMM) is employed to measure the reliability of each tracker. The positions and sizes of predicated bounding boxes of multiple trackers is employed in [33] for trajectory optimization. HMM is employed in [34] to fuse multiple trackers. The confidence score of each tracker is computed with trained parameters. In [35], each tracker is incrementally trained to classify training examples which are not correctly classified in the previous frames. In [39], overlap and distance errors between different ensemble trackers are considered in a short period of time. A history of correlation response is used in [4, 44] to infer the reliability of each independent



Figure 1. Pipeline of our tracking algorithm.

tracker. And the position is estimated by fusing these positions of trackers. A clustering method with temporal constraints is proposed to explore and memorize patterns from previous frames [10].

### 2.3. Backward tracking

Tracking task is decomposed into tracking, learning and detection (TLD) [8] to facilitate each other. The backward tracking is employed to filter effective feature points. Lucas-Kanade method [40] is used to track target and back-tracking is employed in each frame to select the accurate tracking points. Multiple trackers are employed in [9] using different features, e.g., color, texture and illumination invariant features. Each tracker tracks object forwardly and backwardly to measure the reliability of the tracker. In [5], backward tracking is employed to measure the tracking quality. A re-detection scheme is carried out if the tracking quality degrades.

In light of the above observation, we make fully use of backward tracking by evaluating PSR values over a short history of time. Meanwhile, a PSR stability is designed to adaptively fuse different feature tracking results.

## 3. Tracking algorithm

We aim at developing a real-time UAV tracking system that is robust to object appearance changes. Our method based on correlation filter tracking due to its competitive performance and high efficiency. The key idea is to utilize backward tracking with multiple features (e.g., HOG and ColorNames) and develop a PSR stability criterion to adaptively combine the independent trackers. Furthermore, to effectively cope with appearance changes, PSR stability is employed to adaptively update the model. Pipeline of our tracking method is illustrated in Figure 1.

### 3.1. Correlation filter tracking

We tested our tracker with three feature configurations: HOG, ColorNames [31, 36] and HOG with ColorNames. Combination of multiple features can enhance tracking results since these features are complementary to each other. A briefly introduction to the correlation filter tracking algorithm is given here.

Correlation filter tracking makes use of the circulant structure of training and testing examples for speeding up computation efficiency with negligible loss in tracking accuracy [17]. The classifier is trained using an image patch which is centered around the object.

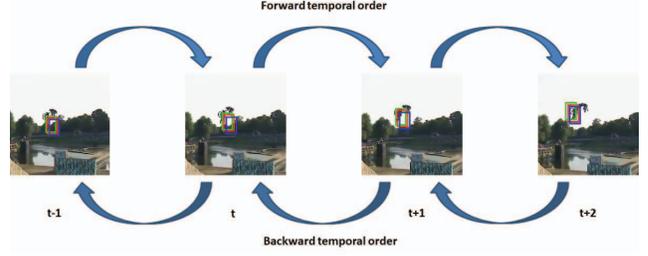


Figure 2. In our approach, we use an inverse temporal order tracking to obtain a reliable tracking by simultaneously exploiting the dual and complementary information from both orders.

Let each sample  $x_k$  contains  $D$  feature channels  $x_j^1, x_j^2, \dots, x_j^D$ ,  $y$  is the Gaussian shape label matrix. The correlation filters algorithm can be formulated as:

$$\operatorname{argmin}_w \sum_{k=1}^K \|y_k - \phi(x_k, w)\|_{L^2}^2 + \lambda \sum_{d=1}^D \|w\|_{L^2}^2, \quad (1)$$

where  $x_k$  is the shifted sample,  $w$  is  $D$  channel features,  $\lambda$  is a regularization parameter.  $\phi(*, *)$  is a mapping function,

$$\phi(x_k, w) = \sum_{d=1}^D x_k^d * w^d, \quad (2)$$

Equation (1) can be computed efficiently in the Fourier domain.  $w = \sum_{k=1}^K \alpha_k x_k$ ,  $\alpha$  can be computed as,

$$\alpha = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(y)}{\mathcal{F}(\phi(x, x)) + \lambda} \right), \quad (3)$$

where  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  represent Fourier transform and its inverse, respectively. Given the appearance model  $\hat{x}$  and  $\alpha$ , the response map  $\hat{y}$  of a patch  $z$  can be computed as follows,

$$\hat{y} = \mathcal{F}^{-1}(\mathcal{F}(\alpha)) \odot \mathcal{F}(\phi(z, \hat{x})), \quad (4)$$

where  $\odot$  is element-wise product.

### 3.2. PSR computation

Our forward and backward tracking is an extension of [5] and is illustrated in Figure 2. The red box in frame  $t$  is the tracking position. The green box and blue box in frame  $t+1$  are positions traced by two different features in forward temporal order. The green box and blue box in frame  $t$  are positions traced by two corresponding features in backward temporal order. The backward tracking is carried out from frame 2. The PSR values are utilized to measure tracking results in each frame. PSR is a measure to quantify the sharpness of the correlation peak. Higher PSR value represents more reliable detection results. Thus, PSR can be used to weight the confidence maps of each feature. PSR is defined as,

$$\operatorname{psr}_i = \frac{\max(\hat{f}_i^t) - \mu_i}{\sigma_i}, \quad (5)$$

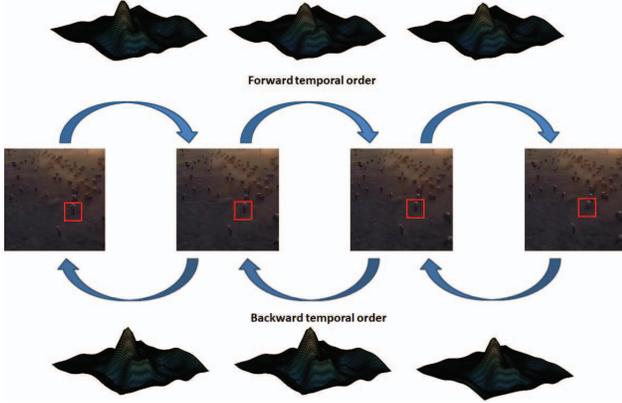


Figure 3. PSR values become larger when the tracking results are accurate.

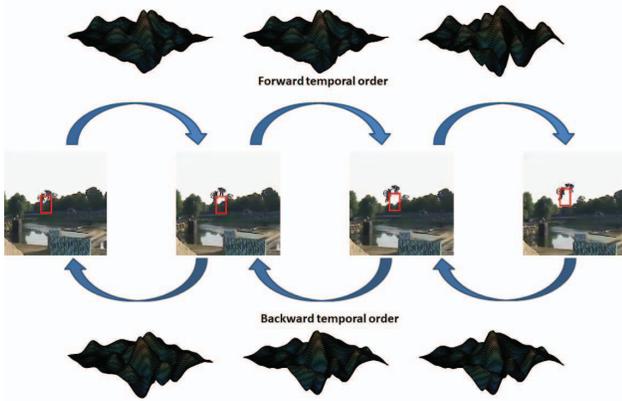


Figure 4. PSR values significantly decrease when tracking drift occurs.

where  $\hat{f}_i^t$  is the confidence map (from equation (4)) of the  $i$ -th feature at time  $t$ .  $\mu_i$  and  $\sigma_i$  represent the mean and standard deviation of the  $i$ -th confidence map respectively. The PSR of both forward and backward tracking are taken into consideration to strengthen the temporal consistency on consecutive frames for reliable tracking results.

Figure 3 and Figure 4 illustrate how the quality of the tracking results improves the response map. In Figure 3, the PSR values are larger because the response map has less noise and sharper peak, which means the tracking results are accurate. The response maps of forward tracking are shown in the top row. The PSR values are 5.929, 5.8075 and 5.6693 respectively. The response maps of backward tracking are shown in the bottom row. The PSR values are 5.5991, 6.1413 and 5.4707 respectively.

In Figure 4, the PSR values become rapidly smaller as tracking drift occurs. The PSR values in the top row are 0.6269, 0.6241 and 0.6240 respectively.

### 3.3. PSR stability

We introduce a PSR stability measure to reveal the distribution of response map. The mean value of PSR over a time period  $T$  is defined as,

$$M_i^t = \frac{1}{T} \sum_{t=1}^T psr_i^t, \quad (6)$$

where  $psr_i^t$  is the PSR value at frame  $t$  of the  $i$ -th feature (as computed by equation (5)). It reveals the temporal trajectory consistency. The fluctuation of the PSR in a short period  $T$  is defined as,

$$V_i^t = \sqrt{\sum_{t=1}^T (M_i^t - psr_i^t)^2}, \quad (7)$$

The PSR stability is then computed as follows,

$$Pw_i^t = \frac{M_i^t}{V_i^t + 0.005}, \quad (8)$$

A PSR value just considers current frame while our PSR stability measure take a period into consideration. Temporal smoothness is useful in tracking for inferring tracking quality. When the fluctuations of PSR values are large, this indicates that tracking drift might have occurred.

### 3.4. Model update

The final position in frame  $t + 1$  can be predicted as follows,

$$hy_{cf} = \beta_1 * hy_1 + \beta_2 * hy_2 + \beta_3 * hy_3, \quad (9)$$

where  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  are the normalized PSR stability of  $Pw_1^t$ ,  $Pw_2^t$  and  $Pw_3^t$  respectively,  $hy_1$ ,  $hy_2$  and  $hy_3$  are the three forward tracking results.

In UAV based tracking, the object appearance changes severely mainly because of illumination variation and occlusion. Thus, it is necessary to update each classifier over time. With correlation filter trackers, the trained model only considers the appearance of the target in the current frame. The tracker updates the classifier coefficients by  $\mathcal{F}(\alpha)^t = (1 - \gamma)\mathcal{F}(\alpha)^{t-1} + \gamma\mathcal{F}(\alpha)$ , where  $\mathcal{F}(\alpha)$  is the classifier coefficient and  $\gamma$  is a learning parameter.

If the tracker uses a fix learning rate, it means that the appearance and correlation filter will be updated without adapting to specific situations. Once a tracker loses the target, the whole model will be corrupted. We solve this problem by adaptively updating the trained model. It is noted that the model of a corrupted feature should not be updated to avoid introducing errors. And the updating frequency should be proportional to the tracking reliability. Similar to PSR stability presented in Section 3.3, a threshold is utilized

Algorithm: tracking algorithm
Input: t-th frame $F_t$ , previous tracking state $s_{t-1}$ .
Output: Find the target location $s_t$ .
1. Extract the target features.
2. Compute the correlation filter response map of each feature with equation (4).
3. Backward tracking with equation (4).
4. Compute the PSR of each feature with equation (5).
5. Compute the PSR stability with equation (8).
6. Predict the object position of the current frame t with equation (9).
7. If $Pw > threshold$
8.     Update the model.
9. Else
10.    Keep the model unchanged.
11. End if

Figure 5. Our tracking algorithm.

to adaptively update each tracker. The learning rate for the model is set proportional to the PSR stability (equation (8)). Thus, the update scheme is defined as,

$$\mathcal{F}(\alpha)_i^t = \begin{cases} (1 - \beta)\mathcal{F}(\alpha)_i^{t-1} + \beta\mathcal{F}(\alpha)_i, & \text{if } Pw_i^t > threshold \\ \mathcal{F}(\alpha)_i^{t-1}, & \text{else} \end{cases} \quad (10)$$

$$x_i^t = \begin{cases} (1 - \beta)x_i^{t-1} + \beta x_i, & \text{if } Pw_i^t > threshold \\ x_i^{t-1}, & \text{else} \end{cases} \quad (11)$$

where  $\beta$  is a parameter, threshold is pre-defined,  $Pw_i^t$  is the PSR weight of the  $i - th$  feature in frame  $t$ .

As showed in Section 3.3, PSR stability infers tracking reliability. The PSR stability is thus taken into consideration in order to avoid unwanted model updating. In our updating scheme, even when the target is occluded in one frame, our scheme can still track the target accurately by using the classifiers of the previous frame. Thus, it is able to predicate the occluded object when it appears again in the next frame. Our tracking algorithm is presented in Figure 5.

## 4. Experiment

### 4.1. Dataset

Previous works are mainly focus on generic tracking datasets, such as OTB [14], LaSOT [15] and TC128 [43]. In recent years, UAV tracking datasets are emerged. To evaluate our method, we test our algorithm on five UAV tracking datasets. There are different challenging attributes in these datasets, such as variations in illumination, occlusion, scale changes, etc. Meanwhile, these UAV tracking datasets focus on different aspects.

The UAV123 dataset [20] contains 123 videos captured by a UAV at 30 fps. There are various tracking situations, e.g., people running on the lawn, cars on the road, and persons with bicycles on the road.

The UAV123-10fps dataset [20] is a down sampled version of the UAV123 dataset.

The UAV20L dataset [20] contains 20 videos with long duration.

The DTB70 dataset [7] contains 70 videos with large displacements.

The VisDrone2018 single object tracking (VisDrone2018-SOT) dataset [41, 46] contains 86 training videos and 11 validation videos. The testing videos are used for competition and ground truths are not released. Thus, these 97 videos (86 training videos and 11 validation videos) can be employed to test our tracking algorithm. The benchmark dataset is captured by various drone-mounted cameras, covering a wide range of attributes including location (taken from different cities), objects (vehicles, pedestrian, bicycles, etc.), environment (urban and country), and density (sparse and crowded scenes). Meanwhile, the dataset is collected using various drone platforms, in different situations, and under various weather and illumination conditions.

### 4.2. Compared trackers

In the UAV123 dataset, UAV123-10fps dataset and UAV20L dataset, we compared our method with six tracking algorithms: ECO [27], Staple [19], BACF [26], SRDCFdecon [29], SRDCF [28] and fDSST [24]. Most of them have achieved state-of-the-art results on OTB dataset [14]. The SRDCF results are provided by [20].

In the DTB70 dataset, we compare our method with six tracking algorithms: ECO [27], Staple [19], BACF [26], SRDCFdecon [29], MEEM [38] and fDSST [24]. The MEEM results are provided by [7].

In the VisDrone2018-SOT dataset, we compare our method with seven tracking algorithms: ECO [27], Staple [19], BACF [26], HDT [4], STRCF [42], CSRDCF [2] and fDSST [24].

These tracking algorithms are recently published and achieve state-of-the-art results in the OTB dataset [14]. Most of them can be run in real-time. We use the source codes released by the authors. The parameters of these methods are fixed and given by the authors.

The evaluation metrics are success and precision rate are used by the OTB dataset [14].

### 4.3. Quantitative evaluation

#### 4.3.1 UAV123 dataset

Our method achieved 0.461 in success plots and 0.660 in precision plots on the UAV123 dataset as shown in Figure 6. The ECO method achieves the first place in both precision and success plots. Our method ranks the third in precision and success plots respectively. The ECO method uses a convolution operator to improve the tracking model. The SRDCF [28] results are provided by the dataset [7]. It performs a slightly better than our method while it runs at 5.24fps [7]. And the speed of our proposed method is 20 fps without code optimization.

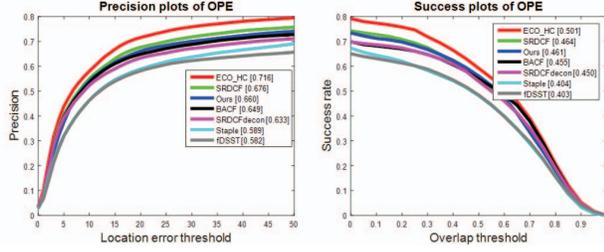


Figure 6. The success plots and precision plots on the UAV123 dataset.

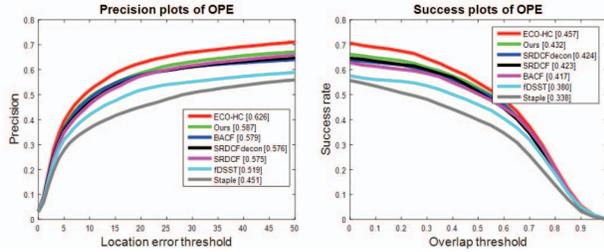


Figure 7. The success plots and precision plots on the UAV123-10fps dataset.

### 4.3.2 UAV123-10fps dataset

Our method achieved 0.432 in success plots and 0.587 in precision plots on the UAV123-10fps dataset as shown in Figure 7. This dataset is used to test the tracker in low frame rate tracking.

All the compared methods drop in precision and success rate compare to the results in the UAV123 dataset. This indicates that the frame rate is an important factor in visual tracking. The SRDCF method drops to the fourth place in the success rate and fifth place in the precision rate. Our method ranks the second place in both precision and success rates. It indicates that our method is suitable in low frame rate tracking which frequently exists in UAV tracking under limited hardware capabilities.

### 4.3.3 UAV20L dataset

Our method achieved 0.400 in success plots and 0.597 in precision plots on the UAV20L dataset as shown in Figure 8. Our method ranks first in precision rate and third in success rate. This dataset is used to test the tracker in long term tracking as the frames of videos are between one thousand and six thousand. Our method achieved competitive performance which indicate that our tracker is effective in long term tracking.

### 4.3.4 DTB70 dataset

Our method achieved 0.407 in success plots and 0.604 in precision plots on the DTB70 dataset as shown in Figure 9.

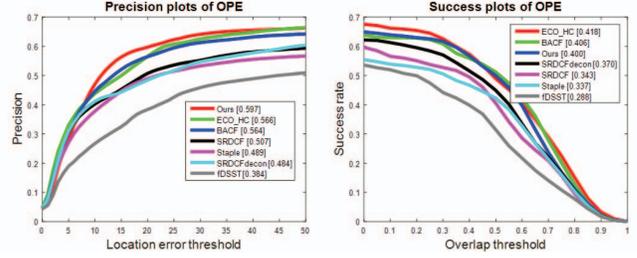


Figure 8. The success plots and precision plots on the UAV20L dataset.

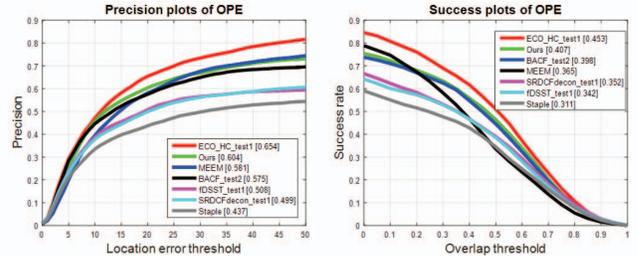


Figure 9. The success plots and precision plots on the DTB70 dataset.

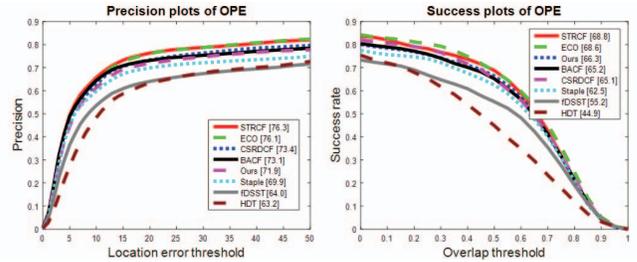


Figure 10. The success plots and precision plots on the VisDrone2018-SOT dataset.

Our method ranks second in both precision rate and success rate. There is severe camera motion in this dataset. Thus, the performance of the compared methods drop compared to the results in the UAV123 dataset.

### 4.3.5 VisDrone2018-SOT dataset

Our method achieved 0.663 in success plots and 0.719 in precision plots on the VisDrone2018-SOT dataset as shown in Figure 10. Our method ranks third in success rate. Our method performs a slightly lower than STRCF and ECO trackers.

## 4.4. Qualitative evaluation

We also plot the results of ECO, BACF, Staple, fDSST and our method for qualitative comparison. Figure 11 shows a qualitative comparison of our method with these trackers on car6 and bike1 videos in the UAV20L dataset. In these two videos, the lengths of the videos are 4861 and



car6



bike1



Figure 11. Qualitative comparison of our method with state-of-the-art methods on the car6 and bike1 videos in the UAV20L dataset.



Soccer2



Surfing03



Figure 12. Qualitative comparison of our method with state-of-the-art methods on the Soccer2 and Surfing03 videos in the DTB70 dataset.

3085 respectively. There are challenging situations in the videos, e.g., viewpoint changes, rotation and scale variations. Our method tracks the targets throughout the entire videos.

Figure 12 shows a qualitative comparison of our method with these trackers on soccer2 and surfing03 videos in the DTB70 dataset. There are occlusion and background clutter in the videos. Our algorithm tracks the targets consistently.

## 4.5. Discussion

In the past few years we have witnessed significant progress in deep neural networks. The popularity of traditional hand-crafted features seems to be overtaken by the deep features, which can learn powerful features automatically from images and have brought breakthroughs in various problems in computer vision. However, these advances rely on deep networks with millions or even billions of parameters, and the availability of GPUs with very high computation capability and large scale labeled datasets plays a key role in their success.

In UAV tracking, computation cost is expensive. Meanwhile, there are only a few UAV tracking dataset and limited amounts of annotated training images can be gathered in UAV tracking.

Our method is based on hand-crafted features and does not to be trained offline. The HOG and color features are easy to be implemented in hardware which can be further accelerated. Moreover, our method is able to achieve appealing results on five UAV datasets which include various challenging factors. The results indicate the robustness and effectiveness of our tracking algorithm. Additionally, it can be run in real-time.

There are two directions to improve our algorithm. First, our PSR stability only considers a short period and each feature independently. Prediction on temporal sequences might be more accurate. Also the correlation between the features should be considered.

Second, more accurate metric can be used to evaluate the correlation response to give a robust measure of tracking.

## 5. Conclusion

Based on the correlation filter tracking framework, we propose a UAV tracking method that uses hand-crafted features. A PSR stability is developed to measure the status of tracking results and estimate fusion weights. By using the PSR stability, our tracker is robust to appearance changes. Our method achieved comparable results with state-of-the-art algorithms on five UAV tracking datasets. Furthermore, our method can operate at 20 fps (on a i7 cpu).

## References

- [1] X. Li, W. Hu, C. Shen, Z. Zhang, A. R. Dick, and A. van den Hengel, A survey of appearance models in visual object tracking, *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 58, 2013.
- [2] Lukezic, Alan, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan, Discriminative correlation filter with channel and spatial reliability, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6309-6318, 2017.

- [3] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, Hierarchical convolutional features for visual tracking, in Proceedings of the IEEE International Conference on Computer Vision, 2015.
- [4] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, Hedged deep tracking, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [5] Yong Wang, Robert Laganriere, Daniel Laroche, Ali Osman Ors, Xiaoyin Xu, Changyun Zhu, Deep Convolutional Correlation Filters for Forward-Backward Visual Tracking, ISVC, pp. 320-331, 2018.
- [6] Zhuojin Sun, Yong Wang, Robert Laganriere, Hard negative mining for correlation filters in visual tracking, Machine Vision Application, 30(3), pp. 487-506, 2019.
- [7] Li, Siyi, and Dit-Yan Yeung, Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models, In AAAI, pp. 4140-4146, 2017.
- [8] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7):1409-1422, 2012.
- [9] Lee, Dae-Youn, Jae-Young Sim, and Chang-Su Kim, Multi-hypothesis trajectory analysis for robust visual tracking, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5088-5096. 2015.
- [10] Wang, Shu, Shaoting Zhang, Wei Liu, and Dimitris N. Metaxas, Visual Tracking with Reliable Memories, In IJCAI, pp. 3491-3497. 2016.
- [11] Z. Yin, F. Porikli and R. Collins, Likelihood Map Fusion for Visual Object Tracking, IEEE Workshop on Applications of Computer Vision (WACV08), Breckenridge, pp.1-7, Jan 2008.
- [12] Zhu, Zheng, Guan Huang, Wei Zou, Dalong Du, and Chang Huang, Uct: Learning unified convolutional networks for real-time visual tracking, In Proceedings of the IEEE International Conference on Computer Vision, pp. 1973-1982. 2017.
- [13] A.W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7):1442-1468, 2014.
- [14] Y. Wu, J. Lim, and M.-H. Yang, Object tracking benchmark, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, pp. 1834-1848, 2015.
- [15] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, LaSOT: A High-quality Benchmark for Large-scale Single Object Tracking, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.
- [16] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, Visual object tracking using adaptive correlation filters, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010.
- [17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in Proceedings of European Conference on Computer Vision, 2012.
- [18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, High-speed tracking with kernelized correlation filters, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583-596, 2015.
- [19] Bertinetto, Luca, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, Staple: Complementary learners for real-time tracking, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1401-1409. 2016.
- [20] Mueller, Matthias, Neil Smith, and Bernard Ghanem, A benchmark and simulator for uav tracking, In European conference on computer vision, pp. 445-461. Springer, Cham, 2016.
- [21] Ma, C., Yang, X., Zhang, C., Yang, M. H, Long-term correlation tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5388-5396, 2015.
- [22] Wang, N., Zhou, W., Li, H, Reliable Re-detection for Long-term Tracking, IEEE Transactions on Circuits and Systems for Video Technology, 2018.
- [23] H. Fan and H. Ling, Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking, IEEE International Conference on Computer Vision (ICCV), 2017.
- [24] Danelljan, Martin, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, Discriminative scale space tracking, IEEE transactions on pattern analysis and machine intelligence 39, no(8), pp. 1561-1575.
- [25] H. Fan and H. Ling, Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.
- [26] Kiani Galoogahi, Hamed, Ashton Fagg, and Simon Lucey, Learning background-aware correlation filters for visual tracking, In Proceedings of the IEEE International Conference on Computer Vision, pp. 1135-1143. 2017.
- [27] Danelljan M, Bhat G, Khan F S, et al. ECO: Efficient Convolution Operators for Tracking. in CVPR 2017.
- [28] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 4310-4318.
- [29] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1430-1438.
- [30] Liu, Ting, Gang Wang, and Qingxiong Yang, Real-time part-based visual tracking via adaptive correlation filters, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4902-4912. 2015.
- [31] Danelljan, Martin, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer, Adaptive color attributes for real-time visual tracking, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090-1097. 2014.

- [32] N.Wang and D.-Y. Yeung, Ensemble-based tracking: Aggregating crowd sourced structured time series data, in Proceedings of the International Conference on Machine Learning, 2014.
- [33] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, Randomized ensemble tracking, in Proceedings of the IEEE International Conference on Computer Vision, 2013.
- [34] T. Vojir, J. Matas, and J. Noskova, Online adaptive hidden markov model for multi-tracker fusion, *Computer Vision and Image Understanding*, vol. 153, pp. 109-119, 2016.
- [35] S. Avidan, Ensemble tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271, 2007.
- [36] J. V. D. Weijer, C. Schmid, J. Verbeek, and D. Larlus, Learning color names for real-world applications, *TIP*, 18(7):1512C1523, 2009.
- [37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.
- [38] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In Proceedings of the European Conference on Computer Vision, 2014.
- [39] Wang, Ning, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li, Multi-cue correlation filters for robust visual tracking, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4844-4853. 2018.
- [40] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221-255, 2004.
- [41] Zhu, Pengfei, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu, Vision meets drones: A challenge, arXiv preprint arXiv:1804.07437, 2018.
- [42] Li, Feng, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang, Learning spatial-temporal regularized correlation filters for visual tracking, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4904-4913. 2018.
- [43] P. Liang, E. Blasch, and H. Ling, Encoding Color Information for Visual Tracking: Algorithms and Benchmark, *IEEE Trans. on Image Processing (T-IP)*, 24(12), 5630-5644, 2015.
- [44] Y Qi, S Zhang, L Qin, Q Huang, H Yao, J Lim, M-H Yang, Hedging Deep Features for Visual Tracking, *IEEE Trans. PAMI*, 2018.
- [45] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian, The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking, *European Conference on Computer Vision (ECCV)*, 2018.
- [46] Wen, Longyin, Pengfei Zhu, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Chenfeng Liu et al, VisDrone-SOT2018: The Vision Meets Drone Single-Object Tracking Challenge Results, In Proceedings of the European Conference on Computer Vision (ECCV), pp. 469-495. 2018.