Published in final edited form as:

Mach Vis Appl. 2019 February ; 30(1): 125–137. doi:10.1007/s00138-018-0981-4.

Robust UAV-based Tracking Using Hybrid Classifiers

Yong Wang,

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

Wei Shi, and

INSKY Lab, Leotail Intelligent Tech, 1628, Pudong District, Shanghai

Shandong Wu

Departments of Radiology, Biomedical Informatics, Bioengineering, and Intelligent Systems,
University of Pittsburgh, USA

Abstract

Visual object tracking is a key factor for unmanned aerial vehicles (UAVs). In this paper, we propose a robust and effective visual object tracking method with an appearance model based on the locally adaptive regression kernel (LARK). The proposed appearance model encodes the geometric structure of the target. The tracking problem is formulated as two binary classifiers via two support vector machines (SVMs) with online model update. The backward tracking which tracks the target in reverse of time is utilized to evaluate the accuracy and robustness of the two SVMs. The final locations are adaptively fused based on the results of the forward tracking and backward tracking validation. Several state-of-the-art tracking algorithms are evaluated on large scale benchmark datasets which include challenging factors such as heavy occlusion, pose variation, illumination variation and motion blur. Experimental results demonstrate that our method achieves appealing performance.

Keywords

UAV-based tracking; locally adaptive regression kernel; forward tracking; backward tracking

1 Introduction

Recent years we have witnessed an explosion in the application of unmanned aerial vehicles (UAVs). Visual tracking based on a UAV has enabled many new applications in computer vision, such as animal monitoring, aerial navigation, obstacle avoidance, crowd monitoring and so on. Although many studies have been done on the aerial tracking, there are still many problems exist caused by the rapid platform motion, image instability and relatively small size of the object. In this paper, we only consider the situation of a single target tracking by a UAV.

Figure 1 shows the differences between generic tracking (i.e., not UAV-based tracking, i.e., Figure1 (c) and (d)) and UAV-based tracking (i.e., Figure1 (a) and (b)). Depending on the

flight altitude and camera orientation, appearance description in UAV-based tracking is more difficult than in generic tracking. Thus, the UAV-based tracking is a challenging problem compared to generic tracking. The tracking algorithms are generally divided into two approaches, generative approaches and discriminative approaches. The generative approaches search around a region and compare the candidates appearance models with templates appearance models. The candidate that achieves the smallest reconstruction error is chosen to be the tracked target. The discriminative methods train a binary classifier to separate the object from background. As the background information is taken into consideration, the discriminative approaches usually achieve better tracking results in challenging situations.

The UAV-based tracking methods usually rely on feature point tracking [8–10, 58] and morphological filtering [1,2,3,4,5,6,7]. The edge information of the target is important in UAV-based tracking since the object is in a smaller size and in perspective view compared in generic tracking. Moreover, the tracking situations with certain challenges are amplified, e.g., full occlusion, abrupt camera motion, scale variation and fast moving objects.

The locally adaptive regression kernel (LARK) captures the most salient cues of the object, such as geometric structure and edges in good detail. Interestingly, in the case of aerial videos, the most observable parts of the target are the edges. Therefore, LARK feature is an appropriate descriptor in the UAV-based visual tracking.

In the discriminative approaches, the classifier relies heavily on the training examples. Model drift may occur when an object changes its appearance. For example, suppose that the target is gradually changing its appearance due to illumination variation. Then the appearance model should be also updated. However, the SVM model is not able to update very frequently to adapt to the appearance changes. Once the model is updated with contaminated samples, tracking drift eventually happens. Visual tracking methods with multiple trackers that improve the tracking performance have been utilized in [20, 33]. Therefore, two SVM models are used in our work. That is, one SVM model (denoted SVM *model1*) stays fixed, and another SVM model (denoted SVM *model2*) is updated online.

When a tracking method is composed of two trackers, it is critical to ensemble the outputs of each trackers to the final results. The fusion scheme can improve the tracking results substantially even when the trackers have high diversity. This part is effective and universal yet it is least exploited in previous literatures. In [32] the forward tracking and backward tracking is utilized to select the correct motion vectors. To evaluate the results of the two SVM models, the backward tracking which locates the target in the reverse order of time is used in our work.

Motivated by the previous work mentioned above, we exploit the description ability of the LARK feature in the aerial visual tracking. Two SVM models are utilized to deal with the challenging scenarios in the aerial tracking. The backward tracking scheme guarantees an adaptive strategy to fuse the tracking results. The proposed tracking algorithm is validated by extensive experiments, where objects are tracked in various situations with a variety of appearance changes. The contributions of this work are three-fold.

- (1) The LARK feature which depicts the geometric structure of the object is integrated into a discriminative tracking method.
- (2) The forward tracking and backward tracking strategy is utilized to adaptively fuse the results of the two SVM models.
- (3) An evaluation of state-of-the-art methods demonstrates the advantages of our tracking method.

This paper represents a significant extension of our preliminary results presented in [47]. In this paper, we provided full derivation of the method with more detailed descriptions. More details in method implementation and data analysis were also incorporated. We added experiments on two widely used datasets for the tracking performance evaluation with comparisons to the state-of-the-art trackers. Furthermore, additional analyses were presented to demonstrate the effectiveness of the proposed tracking algorithm.

The rest of this paper is organized as follows. A brief review of the related work is given in the next section. In Section 3, the main steps of our proposed method are presented in detail. Experiments on benchmark datasets and analysis of the tracking results are conducted in Section 4. Finally, we conclude our work in Section 5.

2 Related work

There are a large number of literatures for visual tracking. For a thorough review, the readers can refer to [24, 25, 26]. Here, we only review the tracking methods that most related to our work.

Many tracking methods focus on target appearance representations which are based on the generative approaches [28, 31, 54, 56, 57] and discriminative approaches [20, 27, 29, 32]. In [28], the Principal Component Analysis (PCA) is employed to decompose the templates into subspace model. In [31], the spatial information is exploited via alignment pooling method. The object appearance representation is based on the local structure appearance model. Nonnegative matrix coding is adopted in [56, 57] to better characterize object appearance and l_2 norm and $l_{1,1}$ norm are added to constraint target appearance, respectively. PageRank algorithm which is used in image retrieval is employed in [54] for object tracking. An adaboost feature selection is employed in [27] to model target appearance. Harr features are employed to model the target appearance and the structure SVM is combined into the track by detection framework [29]. A tracking, learning and detection algorithm is developed in [32]. The three parts collaborate with each other to improve tracking performance. Object is modeled by CIE Lab color space which is insensitive to drastic illumination variation [20]. And multiple SVM model tracking strategy is developed to handle the tracking drift problem. The histories of trackers are saved. The best tracker is chosen based on minimum entropy criterion. The model is then updated by the correct result to prevent tracking drift.

In recent years, correlation filter tracking method is developed [21] and has been attracted a lot attention. The circulant matrix and ridge regression problem are used in [30]. Kernelized correlation filter (KCF) [39] extends the work of [30] and multi-channel HOG feature is introduced. MUSTER [33] is a biology inspired tracking algorithm which includes short

term and long term tracking. The short term approach is correlation filter based with HOG feature and the long term approach is keypoints matching based. SAMF tracker [37] combines color name [38] and HOG feature [15] to model target appearance. And scaling pool is employed to address the scaling variation problem. In DSST [34], feature pyramid is employed to handle the scaling problem. SRDCF [35] utilizes multiple features to model target appearance and introduces a regularization part to penalize correlation filter coefficients according to the spatial position. A similarity metric which is integrated into correlation filter tracking framework is proposed to exploit the relationship of multiple neighbors for object matching [55]. In addition, a large group of negative examples are trained by the correlation filter.

UAV-based tracking. Color histogram is employed in UAV-based tracking [8, 9, 10]. These tracking algorithms rely on the color features. Therefore, they are vulnerable to the situations that the color of the object is similar to the background (e.g. green bicycle and green grass). The morphological filtering method is employed as a feature extraction in [1–4]. To reduce false positives, SVM and the shape descriptor are combined in [5] to produce positive examples. Optical flow is utilized in [6] for aircraft detection. Local features are utilized in [7] to calculate the optical flow. The local features which are updated online are employed to make the tracking algorithm robust to object appearance changing. These generative approaches are prone to drift in complex situations since the background information is not taken into consideration.

Some generic tracking algorithms are tailored for aerial tracking. Multiple instance learning is utilized in [11]. Multiple classifier voting mechanism and multiple strategy resolution representation strategy are used to deal with challenging factors, such as illumination changes and background clutter. Feature point based detection and tracking is developed in [40, 41]. The TLD tracking method is used in [14]. And the experimental results demonstrate that the background which is incorporated into the positive examples leads to tracking drift. In addition, the features utilized in TLD do not consider the UAV-based tracking situation. Therefore, tracker drifts when perspective changing and scale variation occurs. The Struck algorithm is utilized in [12, 13] and achieves well tracking performance. A boosting and pyramid sampling HOG feature extraction method is proposed [58] for vehicle detection and tracking. A hierarchical layered tracking structure is proposed [59] to establish an inter-vehicle mutual assistance mechanism for vehicle tracking. However, the experiments are specific to certain application domains and not diverse enough.

Generic tracking datasets such as CVPR2013 [23], OTB100 [24], VOT16 [43], TC128 [42] and ALOV300 [26] have been widely used to compare with state-of-the-art trackers. We briefly describe these tracking datasets as below.

CVPR2013 and OTB100: CVPR2013 [23] and OTB100 [24] consists of 50 and 100 video sequences, respectively. CVPR2013 is a subset of OTB100. Both of the two datasets are annotated with bounding boxes. There are 11 different factors in the two datasets: low resolution (LR), scale variation (SV), occlusion (OCC), deformation (DEF), fast motion (FM), out-of-view (OV), in-plane rotation (IPR), out-of-plane rotation (OPR), background clutter (BC), illumination variation (IV) and motion blur (MB).

VOT16: VOT16 belongs to visual object tracking challenges (<http://www.votchallenge.net/>). It consists of 25 infrared video sequences and 60 RGB video sequences. The video attributes are similar to OTB100.

Temple-Color 128 (TC128): This dataset consists of 128 video sequences which are designed for evaluating color feature based tracking methods. The bounding boxes in each frame and video attributes are provided.

ALOV300: There are 315 video sequences in the dataset. The data is real-life videos mainly from YouTube with 64 types of targets ranging from a person, human face, an octopus, a ball, a plastic bag, microscopic cells to a can.

VIVID: The VIVID dataset [45] is dedicated to aerial tracking, but it is only 9 sequences.

UAV123: This dataset [44] is recently created for UAV-based tracking. There are 123 video sequences in this dataset. And 115 video sequences is captured by UAV cameras and 8 video sequences rendered by a UAV simulator, which are all annotated with bounding boxes and 12 attributes. The longest video sequence contains 3085 frames and the total length is more than 110k frames. Various scenarios exist in the dataset, e.g., roads, field, buildings, beaches and so on. The tracking objects include person, aerial vehicles, boats, trucks, cars and so on. The tracking situations include background clutter, occlusion, illumination variation, camera motion, viewpoint change, scale variation and so on.

UAV123-10fps: This dataset is the video in UAV123 down sampled to 10fps. It can be used to test the trackers performance in low frame rate tracking.

In this paper, we will test our method on large scale datasets that are used for aerial tracking and compare with state-of-the-art trackers.

3 Tracking algorithm

Our UAV-based tracking method is summarized in Figure. 2. The rectangle bounding box in the initial frame is from the ground truth. The red rectangle box in frame t is the predicted position. The green rectangle boxes are the samples. LARK features are generated and the two SVM classifiers determine the final location via forward and backward tracking. Finally the LARK features of training examples are extracted and fed to SVM *model2*.

3.1 LARK feature

LARK feature has been utilized in object detection [16], salience detection [17], action recognition [19] and face detection [18]. It can encode the structure information of the surrounding pixels even in the presence of data uncertainty and noise. Different from traditional features, LARK feature computes the likelihood of a point to its surrounding pixels based on geodesic distance between points. The distance is the shortest path along a manifold defined by the embedding of the image data in 3D as $\{x_1, x_2, z(x_1, x_2)\}$.

Take an image surface $\mathcal{S}(x_1, x_2) = \{x_1, x_2, z(x_1, x_2)\}$ as an example. The differential arc length is given by $ds^2 = dx_1^2 + dx_2^2 + dz^2$. The derivative of $z(x_1, x_2)$ is

$$dz(x_1, x_2) = \frac{\partial z}{\partial x_1} dx_1 + \frac{\partial z}{\partial x_2} dx_2 = z_{x_1} dx_1 + z_{x_2} dx_2, \quad (1)$$

where z_{x_1}, z_{x_2} are first derivatives along x_1, x_2 respectively. Thus the arc length is computed as follows,

$$ds^2 = dx_1^2 + dx_2^2 + (z_{x_1} dx_1 + z_{x_2} dx_2)^2 = \Delta x^T C \Delta x + \Delta x^T \Delta x, \quad (2)$$

where $x = [dx_1, dx_2]^T$, and $C = \begin{bmatrix} z_{x_1}^2 + 1 & z_{x_1} z_{x_2} \\ z_{x_1} z_{x_2} & z_{x_2}^2 + 1 \end{bmatrix}$ In the LARK calculating procedure, the

pixels are centered around a local window. The window size is small, e.g., 5×5 . That means the $x^T x$ in the local window is trivial. Thus, $ds^2 \approx x^T C x$. The LARK feature is represented as a self-similarity between a center pixel and its surrounding pixels:

$$K(C_l, \Delta x_l) = \exp(-ds^2) = \exp(-\Delta x_l^T C_l \Delta x_l), \quad (3)$$

where $l \in [1, \dots, P]$, P is the number of pixels centered around the central pixel. C is calculated by the eigen-decomposition as follows,

$$C_l = (s_1 s_2 + \epsilon)^\alpha \left(\frac{s_1 + \tau}{s_2 + \tau} u_1^T u_1 + \frac{s_2 + \tau}{s_1 + \tau} u_2^T u_2 \right), \quad (4)$$

where s_1, s_2 are singular values, u_1, u_2 are eigenvectors of C_l , ϵ, τ, α are set to $10^{-7}, 1, 0.5$ respectively.

LARK feature which is similar to SIFT feature and HOG feature calculates the oriented gradients. In SIFT and HOG, quantization is utilized to oriented gradients. While in LARK feature, eigen-decomposition is able to compute the oriented gradients accurately. Meanwhile, the geodesic distance is treated as a similarity measure between the two pixels, which provides a reliable distance metric and explains the promising results achieved by the previous methods. Figure. 3 illustrates the LARK kernel. A darker blue color represents a bigger value. The key idea behind the LARK feature is to robustly and effectively restore local geometric structures by calculating the differences based on estimated gradients, which depicts the orientation information of the object. Finally, the LARK kernel calculated in each pixel is concatenated to a feature vector to represent the target.

3.2 SVM models

The standard SVM with hinge loss and L_2 regularization [46] is employed in our work. During tracking, two SVM models are utilized to estimate object location by searching for

the maximum classification score around the position from the previous frame. SVM *model1* is only trained in the initial frame and not updated during tracking. Our observation is that the initial bounding box is the target. All the other tracked positions are predicated by the tracker and noise may be involved. If the training samples are contaminated by noise, model drift may occur. For example, when a vehicle goes through a tunnel, the appearance will be changed by illumination variation. And the appearance will be recovered once the vehicle goes out of the tunnel. In the visual tracking, there are many similar situations (e.g., occlusion, pose variation) that the object will be recovered to the initial appearance. Therefore, if the SVM model updates too frequently, tracking results will be degraded. Thus a conservative model update scheme is adopted. Given the predicated location in the current frame, a group of training examples are collected which is utilized to update the SVM *model2* online.

3.3 Forward and backward tracking

Here, a detailed description of our forward and backward tracking strategy is given. The critical idea of our strategy is to find an objective criterion to evaluate the accuracy of the tracking results. Theoretically, a robust tracking algorithm should be identical to the ground truth during forward and backward tracking. Thus, forward and backward tracking is employed to evaluate the robustness of the two trackers. The backward tracking is initialized in the second frame. By comparing the backward trajectories with the tracked positions, we can approximately measure the accuracy of the forward trackers. The best forward position can be determined according to the forward and backward comparison. The final tracking results are more accurate and robust than single model tracking.

Figure. 4 shows two trackers from frame $t-1$ to frame t . The object is tracked within the red rectangle box at frame $t-1$. Black and blue rectangle boxes represent the two forward tracking results at frame t . Green and yellow rectangle boxes denote the corresponding two backward tracking results at frame $t-1$ by SVM *model1* and SVM *model2*, respectively. Comparing the intersection over union (IoU) between the green rectangle box and red rectangle box in frame $t-1$, we can tell approximately how much the SVM *model1* matches the object appearance. In the same way, comparing the IoU between the yellow rectangle box and red rectangle box, we can evaluate the SVM *model2*. The IoU can be utilized to be a fusion weight in the forward tracking. The IoU is calculated as follows [46]:

$$\alpha_i = \frac{SVMB_i \cap PB}{SVMB_i \cup PB}, i = 1, 2, \quad (5)$$

where $SVMB_1$ and $SVMB_2$ represent the SVM *model1* and SVM *model2* tracking rectangle box in frame t respectively. And PB denotes the final tracking box in frame t . U and $\cap$ are intersection and union of the two boxes, respectively. In our tracking algorithm, the final location is determined by the black and blue rectangle boxes.

$$Y = \beta_1 * y_1 + \beta_2 * y_2, \quad (6)$$

where β_1 and β_2 denote the normalized IoU α_1 and α_2 respectively, y_1 and y_2 represent two forward tracking results. The tracking algorithm is illustrated in Figure 5.

4 Experiments

4.1 Dataset

We test our tracker on UAV123 dataset and UAV123-10fps dataset. We also compare these trackers in 12 attributes as shown in Table 1.

To test the generality of our tracker, we further test our method on CVPR2013 dataset [23].

4.2 Compared trackers

The compared tracking algorithms in UAV123 and UAV123-10fps are: DCF [39], KCF [39], IVT [28], TLD [32], Struck [29], OAB [27], CSK [30], ASLA [31], MEEM [20], MUSTER [33], DSST [34], SRDCF [35], SAMF [37] and MOSSE [21]. The results of these methods are all from <https://ivul.kaust.edu.sa/Pages/Dataset-UAV123.aspx>.

The compared tracking methods in CVPR2013 are, e.g., ASLA [31], CSK [30], CXT [49], LSK [50], MIL [48], SCM [53], Struck [29], TLD [32], VTD [52], VTS [51] and so on.

4.3 Quantitative evaluation

We follow the evaluation method in [23]. Two criteria are used to compare the methods: precision and success. Precision is computed as the distance between the centers of a tracker bounding box and the ground truth bounding box. Success is computed by the IoU of pixels between a tracker bounding box and the ground truth bounding box. The precision plot denotes the percentage tracker bounding boxes within a given threshold distance in pixels of the ground truth. To rank the methods, a threshold of 20 pixels [23] is used. The success plot denotes the percentage of tracker bounding boxes whose overlap score is larger than a given threshold. The trackers are ranked using the area under the curve (AUC) measure [23].

One-pass evaluation (OPE) demonstrates how well the bounding box of a tracking method in all frames given the first frame bounding box.

4.3.1 Evaluation on UAV123 dataset—Overall performance: Figure 6 illustrates our tracking method achieves 0.644 and 0.402 in precision and success plots of OPE on UAV123 dataset respectively. The precision plots of video sequences with different factors on UAV123 dataset are illustrated in supplemental file (Figure 1). The number in the title indicates the number of videos. The success plots of video sequences with different factors on UAV123 dataset are illustrated in supplemental file (Figure 2). The number in the title indicates the number of videos. Attribute based performance: Among existing algorithms, our tracker performs well with overall success in aspect ratio change (0.347), scale variation (0.435), low resolution (0.289), full occlusion (0.242), fast motion (0.28), out-of-view

(0.352), partial occlusion (0.342), illumination variation (0.319), background clutter (0.297), viewpoint change (0.373), similar object (0.405) and camera motion (0.401). In most of the scenarios, our tracker yields a favorable tracking results compared to the other compared methods.

4.3.2 Evaluation on UAV123-10fps dataset—Figure 7 illustrates UAV123-10fps dataset. Our tracker achieves 0.608 and 0.391 in precision and success plots of OPE respectively. The precision plots of videos with different factors on UAV123-10fps dataset are illustrated in supplemental file (Figure 3). The success plots of videos with different factors on UAV123-10fps dataset are illustrated in supplemental file (Figure 4).

Attribute based performance: Among existing methods, our method performs well with overall success in scale variation (0.349), aspect ratio change (0.342), low resolution (0.255), fast motion (0.227), full occlusion (0.223), partial occlusion (0.349), out-of-view (0.317), background clutter (0.332), illumination variation (0.340), viewpoint change (0.357), camera motion (0.372) and similar object (0.408). In most of the situations, our method yields a favorable tracking performance compared to the other compared trackers.

4.3.3 Evaluation on CVPR2013 dataset—Figure 8 illustrates CVPR2013 dataset. Our tracker achieves 0.787 and 0.543 in precision and success plots of OPE respectively. The precision plots of videos with different factors on CVPR2013 dataset are illustrated in supplemental file (Figure 5). The success plots of videos with different factors on CVPR2013 dataset are illustrated in supplemental file (Figure 6).

Attribute based performance: Among existing methods, our method performs well with overall success in illumination variation (0.485), out-of-plane rotation (0.531), scale variation (0.518), occlusion (0.537), deformation (0.566), motion blur (0.474), fast motion (0.514), in-plane rotation (0.483), out-of-view (0.572), background clutter (0.553) and low resolution (0.375).

4.4 Qualitative evaluation

Figure 9 demonstrates sampled results of representative video sequences where the objects undergo challenging situations (e.g., low resolution, pose variation, illumination variation). In the bike1 video sequence, a person undergoes rapid appearance changes caused by the illumination variations and viewpoint changes. Our SVM *model2* is updated online and adapt to the appearance changes. The forward tracking and backward tracking strategy adaptively fuses the tracking results to track the target accurately. The KCF method and struck method lock to the background. The MUSTER method does not handle the scale changes well. In the boat1 video sequence, the object undergoes appearance variation due to scale changes. Our tracker is able to locate the boat through the entire video sequence. In the group1 video sequence, a person walks while the appearance changes much caused by illumination variations and low resolution. Our method performs well as the two SVM models adapt to the appearance changes. The KCF method locks to another person. In wakeboard8 video sequence, the object undergoes significant appearance changes due to rotation, occlusion and low resolution. The DSST, MUSTER, KCF and SAMF trackers lock on to the background. The proposed feature encodes the edge information of the object and

tracks the target well. Figure 10 presents some key frames with bounding boxes reported by 10 compared trackers for four videos. The Struck, CXT and MIL methods lose track of the target gradually due to significant deformation and rotation in basketball sequence. Our method tracks the target well due to the LARK feature captures the object representation accurately. Bolt sequence illustrates the performance of these methods in dealing scenarios with background clutters, viewpoint change and deformation, only our tracker and VTD succeed throughout the entire video. Our method handles the challenging situations by the aid of adaptively model update which adapts to the appearance changes. When obvious occlusion occurs as in jogging-2, only TLD and our method track the target well. Our backward tracking scheme decreases the weight of the occlusion model which is able to alleviate the partial occlusion problem. The TLD, VTD, CXT, VTS, CSK, CT, LSK and MIL are not able to keep tracking the object after occlusion in woman video. Only our method and Struck track the target throughout the whole video. Overall, our method is able to deal with these challenges well to alleviate model drift.

4.5 Computational cost

Run time of our tracking algorithm grows proportionally as the number of samples and window size (in section 3.1). Table 2 shows the average per-frame runtime of our algorithm in different number of samples and window size. At present, our tracking algorithm can not run in real time. The main computational cost is in feature computation. While LARK feature can be computed parallel according to section 3.1.

4.6 Discussion

In general, our tracker is among the top three performers in terms of success and precision plot. Since the LARK feature encode the edge feature of the object, our tracking results perform better in low resolution. The forward and backward tracking scheme prevents the model drift in challenge scenarios, such as occlusion and viewpoint change. The most difficult attributes seem to be background clutter and similar object (as shown in supplemental file Figure 1 and Figure 2).

There is still much room for improvement especially for these attributes. For example, the model update can be improved by integrating more adaptive updating scheme. And other classifiers, such as structure SVM, random forest can be considered.

Although our tracker shows appealing performance in the two datasets, there is still much room for improvement. We think there are three possible solutions here:

- (1) The model update can be improved by integrating more adaptive updating strategy.
- (2) The structure SVM is a more advanced classifier that can be taken into consideration.
- (3) The LARK kernel can be more accurate to depict the target by introducing more motion information.

However, these improvements are beyond the scope of this paper, we would like to study them in our future work.

5 Conclusion

In this paper, we have proposed a robust UAV-based visual tracking framework. The discriminative tracking method is achieved by first making use of the LARK feature to depict the edge information of the object. Next we utilize two SVM classifiers to predicate the object position. The forward tracking and backward tracking strategy which is used to measure the tracking accuracy of the two classifiers guarantees the efficiency and robustness of the tracking results in various situations. The final location is determined by the two SVM tracking models. The positive and negative examples are utilized to train and update the SVM *model2* online to adapt to appearance changes. Experimental results have demonstrated that the proposed tracking algorithm outperforms most of the existing state-of-art methods in terms of accuracy and robustness.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This work was partially supported by a National Institutes of Health (NIH) / National Cancer Institute (NCI) R01 grant (#1R01CA193603).

References

1. Wainwright A and Ford J, Fusion of morphological images for airborne target detection, in Information Fusion, 15th International Conference on, pp. 1180–1187, 2012.
2. Gandhi T, Yang M-T, Kasturi R, Camps O, Coraor L, and Mc-Candless J, Detection of obstacles in the flight path of an aircraft, IEEE CVPR, vol. 2, pp. 304–311, 2000.
3. Carnie R, Walker R, and Corke P, Image processing algorithms for uav sense and avoid, ICRA, pp. 2848–2853, 2006.
4. Lai JS, Ford JJ, Mejias L, OShea PJ, and Walker RA, See and avoid using on-board computer vision, in Sense and Avoid in UAS, Research and Applications, John Wiley Sons, pp. 265–294, 2012.
5. Dey D, Geyer C, Singh S, and Digioia M, Passive, long-range detection of aircraft: Towards a field deployable sense and avoid system, in Field and Service Robotics, ser. Springer Tracts in Advanced Robotics, vol. 62, pp. 113–123, 2010.
6. Mccandless JW, Detection of aircraft in video sequences using a predictive optical flow algorithm, pp. 523–530, 1999.
7. Mian A, Realtime visual tracking of aircrafts, in Digital Image Computing: Techniques and Applications, pp. 351–356, 2008.
8. Mondragon IF, Campoy P, Olivares-Mendez MA, and Martinez C, 3D object following based on visual information for Unmanned Aerial Vehicles, in IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, pp. 1–7, 2011.
9. Teuliere C, Eck L, and Marchand E, Chasing a moving target from a flying UAV, in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE, September, pp. 4929–4934, 2011.
10. Kendall A, Salvapantula N, and Stol K, On-board object tracking control of a quadcopter with monocular vision, in Unmanned Aircraft Systems, International Conference on, pp. 404–411, 2014.
11. Fu C, Carrio A, Olivares-Mendez M, Suarez-Fernandez R, Campoy P, Robust real-time vision-based aircraft tracking from unmanned aerial vehicles, IEEE ICRA, pp. 5441–5446, 2014.

12. Lim H, Sinha SN, Monocular localization of a moving person onboard a quadrotor mav, IEEE ICRA, pp. 2182–2189, 2015.
13. Mueller M, Sharma G, Smith N, Ghanem B, Persistent aerial tracking system for uavs, IEEE IROS, pp. 1562–1569, 2016.
14. Pestana J, Sanchez-Lopez J, Campoy P, Saripalli S, Vision based gps-denied object tracking and following for unmanned aerial vehicles, In : Safety, Security, and Rescue Robotics, IEEE International Symposium on, pp. 1–6, 2013
15. Dalal N and Triggs B, Histogram of oriented gradient for human detection, IEEE CVPR, pp. 886–893, 2005.
16. Seo H and Milanfar P, Training-free, generic object detection using locally adaptive regression kernels, IEEE TPAMI, 32(9), pp. 1688–1704, 2010.
17. Seo HJ and Milanfar P, Static and space-time visual saliency detection by self-resemblance, Journal of Vision, 9(12), pp. 1–27, 2009.
18. Seo HJ and Milanfar P, Face verification using the lark representation, IEEE Transactions on Information Forensics and Security, 6(4), pp. 1275–1286, 2011.
19. Seo HJ and Milanfar P, Action Recognition from One Example, IEEE TPAMI, 33(5), pp. 867–882, 5 2011.
20. Zhang Jianming, Ma Shugao, and Sclaroff Stan, MEEM: robust tracking via multiple experts using entropy minimization, ECCV, pp. 188–203, 2014.
21. Bolme DS, Beveridge JR, Draper BA, Lui YM, Visual object tracking using adaptive correlation filters, IEEE CVPR, pp. 2544–550, 2010.
22. Wang Z, Vucetic S, Online training on a budget of support vector machines using twin prototypes, Statistical Analysis and Data Mining 3(3), pp. 149–169, 2010.
23. Wu Y, Lim J, Yang M H, Online object tracking: A benchmark, IEEE CVPR, pp. 2411–2418, 2013.
24. Wu Yi, Lim Jongwoo, and Yang Ming-Hsuan, Object tracking benchmark, IEEE TPAMI 37(9), pp. 1834–1848, 2015.
25. Yilmaz A, Javed O, and Shah M, Object tracking: A survey, ACM Computing Surveys, 38(4), 2006.
26. Smeulders AWM, Chu DM, Cucchiara R, Calderara S, Dehghan A, and Shah M, Visual tracking: An experimental survey, IEEE TPAMI, 36(7), pp. 1442–1468, 2014.
27. Grabner H, Grabner M, Bischof H, Real-time tracking via on-line boosting, In BMVA, (6.1–6.10), 2006.
28. Ross D, Lim J, Lin RS, Yang MH, Incremental learning for robust visual tracking, IJCV, 77(1), pp. 125–141, 2008.
29. Hare Sam, Golodetz Stuart, Saffari Amir, Vineet Vibhav, Cheng Ming-Ming, Hicks Stephen L., and Philip HS Torr, Struck: Structured output tracking with kernels, IEEE TPAMI 38(10), pp. 2096–2109, 2016.
30. Henriques J, Caseiro R, Martins P, Batista J, Exploiting the circulant structure of tracking-by-detection with kernels, ECCV, pp. 702–715, 2012.
31. Jia X, Lu H, Yang MH, Visual tracking via adaptive structural local sparse appearance model, IEEE CVPR, pp. 1822–1829, 2012.
32. Kalal Z, Mikolajczyk K, Matas J, Tracking-Learning-Detection, IEEE TPAMI 34(7), pp. 1409–1422, 2011.
33. Hong Z, Chen Z, Wang C, Mei X, Prokhorov D, Tao D, Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking, IEEE CVPR, pp. 749–758, 2015.
34. Danelljan M, Hager G, Khan FS, and Felsberg M, Accurate scale estimation for robust visual tracking, in BMVC, 2014.
35. Danelljan Martin, Hager Gustav, Fahad Shahbaz Khan, and Michael Felsberg, Learning spatially regularized correlation filters for visual tracking, IEEE ICCV, pp. 4310–4318, 2015.
36. Collins R, Zhou X, and Teh SK, An open source tracking testbed and evaluation web site, IEEE Intl Workshop on PETS, pp. 17–24, 2005.

37. Li Y and Zhu J, A scale adaptive kernel correlation filter tracker with feature integration, IEEE ECCVW, pp. 254–265, 2014.
38. van de Weijer J, Schmid C, Verbeek JJ, and Larlus D, Learning color names for real-world applications, IEEE TIP, 18(7), pp. 1512–1524, 2009.
39. Henriques JF, Caseiro R, Martins P, Batista J, High-speed tracking with kernelized correlation filters, IEEE TPAMI, pp. 583–596, 2015.
40. Nussberger A, Grabner H, Van Gool L, Aerial object tracking from an airborne platform, In: Unmanned Aircraft Systems, International Conference on, pp. 1284–1293, 2014.
41. Qadir A, Neubert J, Semke W, Schultz R chap, On-Board Visual Tracking With Unmanned Aircraft System (UAS). Infotech@Aerospace Conferences, AIAA, 2011.
42. Liang Pengpeng, Blasch Erik, and Ling Haibin, Encoding color information for visual tracking: Algorithms and benchmark, IEEE TIP 24(12), pp. 5630–5644, 2015.
43. Kristan M, Pflugfelder R, Leonardis A, Matas J, Porikli F, Cehovin L, Nebelha G, Fernandez G, Vojir T, Gatt A and Khajenezhad A, The VOT2013 challenge results, In CVPRW, pp. 98–111, 2013.
44. Mueller Matthias, Smith Neil, and Ghanem Bernard, A benchmark and simulator for uav tracking, ECCV, pp. 445–461, 2016.
45. Collins R, Zhou X, and The SK, The PASCAL VOC2010 Results, 2010.
46. Everingham M, Van Gool L, Williams CKI, Winn J, and Zisserman A, The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010.
47. Wang Yong, Shi Wei, and Wu Shandong, Robust UAV-Based Tracking Using Hybrid Classifiers, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, pp. 2129–2137. 2017.
48. Babenko Boris, Yang Ming-Hsuan, and Belongie Serge, Robust Object Tracking with Online Multiple Instance Learning, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8), 1619–1632, 2011. [PubMed: 21173445]
49. Dinh TB, Vo N, and Medioni G, Context Tracker: Exploring supporters and distracters in unconstrained environments, In CVPR, 2011.
50. Liu B, Huang J, Yang L, and Kulikowsk C, Robust Tracking using Local Sparse Appearance Model and K-Selection, In CVPR, 2011.
51. Kwon J and Lee KM, Tracking by Sampling Trackers, In ICCV, 2011.
52. Kwon J and Lee KM, Visual Tracking Decomposition, In CVPR, 2010.
53. Zhong W, Lu H, and Yang M-H, Robust Object Tracking via Sparsity-based Collaborative Model, In CVPR, 2012.
54. Gong Chen, Fu Keren, Loza Artur, Wu Qiang, Liu Jia, Yang Jie, PageRank Tracker: From ranking to tracking, IEEE Transactions on Cybernetics (TCYB), 2013.
55. Liu F, Gong C, Huang X, Zhou T, Yang J, Tao D, Robust visual tracking revisited: From correlation filter to template matching, IEEE Transactions on Image Processing 27 (6), 2777–2790.
56. Liu Fanghui, Gong Chen, Zhou Tao, Fu Keren and Yang Jie, Visual Tracking via Non-negative Multiple Coding, IEEE Transactions on Multimedia (TMM), 2017.
57. Liu Fanghui, Zhou Tao, Gong Chen, Fu Keren, Bai Li, and Yang Jie, Inverse Nonnegative Local Coordinate Factorization for Visual Tracking, IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2017.
58. Cao X, Wu C, Lan J, Yan P, Li X, Vehicle detection and motion analysis in low-altitude airborne video under urban environment, IEEE Transactions on Circuits and Systems for Video Technology, 21(10), 1522–1533, 2011.
59. Cao X, Jiang X, Li X, Yan P, Correlation-based tracking of multiple targets with hierarchical layered structure, IEEE transactions on cybernetics, 48(1), 90–102, 2018. [PubMed: 27875236]

Fig. 1. Differences between UAV-based tracking and generic tracking. (a) and (b) are from UAV-based tracking sequence captured by ourselves. (c) and (d) are from generic tracking dataset OTB100 [24].

Fig. 2.
The proposed UAV-based tracking framework.

Fig. 3.
Examples of normalized LARK value.

Fig. 4.
The forward and backward tracking.

Author Manuscript	Input: Current position at frame t . SVM model1 and SVM model2.
	<ol style="list-style-type: none"> 1. Generate particles within the particle filtering framework. 2. Compute LARK feature for each of the particles and concatenate to a feature vector. 3. Estimate positions by SVM model1 and SVM model2 respectively. 4. Backward tracking by SVM model1 and SVM model2 respectively. And compute the overlap ratios by equation (5). 5. Compute the final position by equation (6). 6. Generate positive samples and negative samples. And update SVM model2.
	Output: Tracked target position at frame $t+1$. SVM model1 and updated SVM model2.

Fig. 5.
Our tracking algorithm.

Fig. 6.
Precision and success plots of OPE on UAV123 dataset. “LARK_SVM” represents our method.

Fig. 7.

Precision and success plots of OPE on UAV123-10fps dataset. “LARK_SVM” represents our method.

Fig. 8.
Precision and success plots of OPE on CVPR2013 dataset. “LARK_SVM” represents our method.

Fig. 9. Qualitative results of the 8 trackers over sequences bike1, boat1, group1, wakeboard8, in which the targets undergo severe appearance changes.

Fig. 10. Qualitative results of the 10 trackers over sequences basketball, bolt, jogging-2, woman, in which the targets undergo severe appearance changes.

Table 1

Attributes used in UAV123 dataset.

Attribute Name	Property
Aspect ratio change	The fraction of ground truth aspect ratio at least one frame is outside the range [0.5, 2].
Background Clutter	The background near the object has similar appearance as the object.
Camera Motion	Abrupt motion of the camera.
Fast Motion	Motion of the ground truth box is larger than 20 pixels between two consecutive frames.
Full Occlusion	The object is fully occluded.
Illumination Variation	The illumination of the object changes significantly.
Low Resolution	At least one ground truth box has small than 400 pixels.
Out-of-View	Some portion of the object leaves the view.
Partial Occlusion	The object is partially occluded.
Similar Object	There are distractors of similar shape or same type near the object.
Scale Variation	The ratio of initial and at least one subsequent bounding box is outside the range [0.5, 2].
Viewpoint Change	Viewpoint affects object appearance significantly.

Table 2

Average per-frame runtime of our algorithm.

samples	3 * 3	5 * 5
100	0.302	1.024
200	0.600	1.964
300	0.872	3.002
400	1.153	3.986
500	1.455	4.975
600	1.902	6.122