Robust UAV-based Tracking Using Hybrid Classifiers

Yong Wang
University of Ottawa
75 Laurier Ave E, Ottawa, ON K1N 6N5
ywang6@uottawa.ca

Wei Shi
INSKY Lab, Leotail Intelligent Tech
1628, Pudong District, Shanghai
weishiInsky@126.com

Shandong Wu
Department of Radiology, Biomedical Informatics, and Bioengineering, University of Pittsburgh, USA
3362 Fifth Avenue Room 130, Pittsburgh, PA, 15213, USA
wus3@upmc.edu

Abstract

Robust object tracking plays an important role for unmanned aerial vehicles (UAVs). In this paper, we present a robust and efficient visual object tracking algorithm with an appearance model based on the locally adaptive regression kernel (LARK). The proposed appearance model preserves the geometric structure of the object. The tracking task is formulated as two binary classifiers via two support vector machines (SVMs) with online update. The backward tracking which tracks the object in reverse of time is employed to measure the accuracy and robustness of the two trackers. The final positions are adaptively fused based on the results of the forward tracking and backward tracking validation. Several state-of-the-art trackers are evaluated on the UAV123 benchmark dataset which includes challenging situations such as illumination variation, motion blur, pose variation and heavy occlusion.

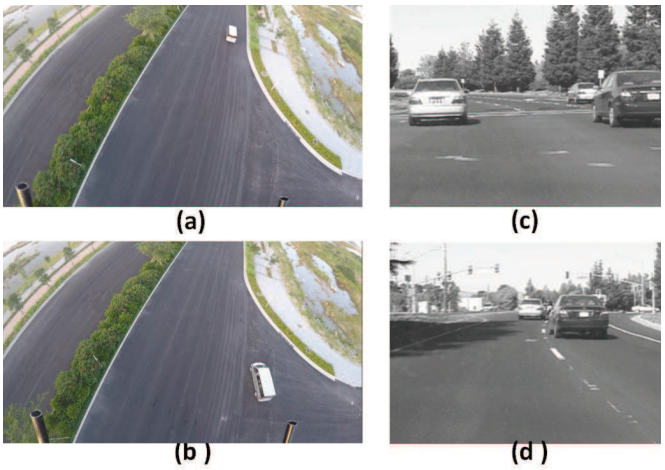

Figure 1. Differences between UAV-based tracking and generic tracking. (a) and (b) are from UAV-based tracking sequence captured by ourselves. (c) and (d) are from generic tracking dataset OTB100 [24].

1. Introduction

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

Figure 1 illustrates the differences between UAV-based tracking and generic tracking (i.e., not UAV-based tracking, such as Figure 1 (c) and (d)). Depending on the camera orientation and flight altitude, appearance model in UAV-based tracking is not as rich as generic tracking. Thus, the UAV-based tracking is a challenging task compared to generic tracking. The tracking methods are generally categorized into two approaches, generative tracking methods and discriminative tracking methods. The generative methods search around an area and compare the templates appearance models with candidate models. The candidate which achieves the smallest reconstruction error with the templates is chosen to be the target. The discriminative approaches train a binary classifier to separate the target from background. Since the background information is taken into account, the discriminative methods usually achieve better performance in challenging scenarios.

The UAV-based tracking methods rely on morphological filtering [1–7] and feature point tracking [8–10]. The edge

of the object is important as the target is in perspective view and small size compared in generic tracking. Furthermore, the tracking scenarios with certain challenges are amplified, e.g., fast moving objects, full occlusion, scale variation and abrupt camera motion.

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

In the discriminative tracking, the classifier model relies heavily on the training samples. Tracking drift may occur when a target changes its appearance. For example, suppose that an object is gradually changing its appearance by illumination variation. Then the appearance model is also updated. However, the SVM model is difficult to update very frequently to adapt to the appearance changes. Once the appearance model is updated with noisy and misalignment training samples, tracking drift eventually happens. Visual object tracking algorithms with multiple trackers which improve the tracking results have been used in [21,33]. Therefore, two SVM models are employed in our work. That is, SVM *model1* stays fixed, and SVM *model2* is updated online.

When a tracking system consists of two trackers, it is important to ensemble the outputs of the individual trackers to the final result. The results fusion strategy can improve the tracking performance substantially especially when the trackers have high diversity. This part is universal and effective yet it is least explored in previous works. In [32] the forward and backward tracking is used to select the correct motion vectors. To adaptively fuse the tracking results of the two SVM models, the backward tracking which tracks an object in the reverse order of time is employed in our work.

Motivated by the work mentioned above, in this paper, we exploit the LARK feature for the aerial visual tracking. And integrate the feature into a discriminative UAV tracking system. Two SVM models are employed to handle the challenging factors in the UAV-based tracking. The backward tracking provides an adaptive scheme to fuse the tracking results. This tracking system provides an accurate and robust solution for aerial situations. The proposed system is validated by extensive experiments, where targets are tracked in various scenarios with a variety of scale and appearance changes. The contributions of this paper are three fold.

(1) The LARK feature which encodes the geometric structure of the targets is integrated to the discriminative tracking framework.

(2) The forward and backward tracking scheme is employed to adaptively fuse the tracking results obtained by two SVM models.

(3) An evaluation of state-of-the-art trackers illustrates the advantages of our tracking algorithm.

2. Related work

There are a plethora of literatures for visual object tracking. For a thorough survey, the readers can refer to [24–26]. Here, we only focus on the tracking algorithms most related to our work.

Many tracking algorithms focus on objet appearance representations. An online adaboost feature selection is used in [27] to adaptively model object appearance. Template is decomposed into subspace model by Principal Component Analysis (PCA) and updated adaptively during tracking [28]. In [29], Harr features are used to model the object and the structure output SVM is integrated into the track by detection framework. In [31], the authors exploit the spatial information based on alignment pooling method. The appearance representation is based on the local structure sparse appearance model. A tracking, learning and detection framework is proposed in [32]. The three components cooperate with each other to improve tracking results. Target is modeled by CIE Lab color space which is robust to drastic illumination variation and multiple SVM trackers tracking scheme is proposed in [20] to address the model drift problem. The history snapshots of trackers are restored. The best tracker in current frame is selected based on minimum entropy criterion. The model is then updated by the correct tracking data to prevent model drift.

Recently, correlation filter based tracking method is introduced [21] and has been paid a lot attention. The ridge regression problem and circulant matrix are employed in [30] to be effectively kernelized. MUSTER [33] is a biology inspired tracking method which combined short term and long term tracking. The short term tracking is correlation filter based with HOG feature [15] and the long term tracking is keypoints matching based. SAMF [37] integrate color name [38] and HOG feature to model object appearance. And scaling pool is used to handle the scaling variation problem. KCF [39] extends the work of CSK [30] and introduce multi-channel HOG feature. Feature pyramid is used in DSST [34] to address the scaling problem. SRDCF [35] uses multiple features to model object appearance and introduces a spatial regularization component to penalize correlation filter coefficients according to the spatial location. A larger set of negative training samples are learned by the correlation filter.

Color histogram is utilized in aerial tracking [8–10]. These tracking methods rely on the color feature. Thus, they are vulnerable to the scenarios that the color of the target is similar to the background (e.g. gray car and gray lanes). The morphological filtering technology is used as a basic feature extraction in [1–4]. In order to reduce false

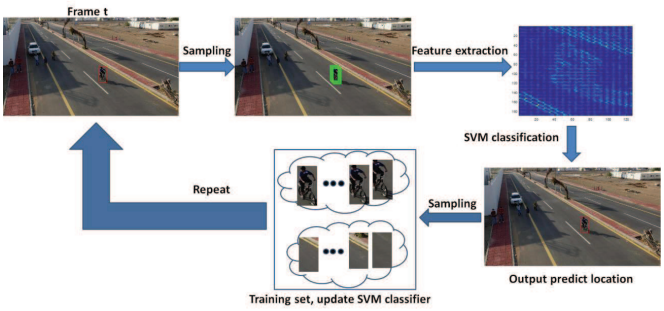

Figure 2. Our UAV-based tracking framework.

positives, the shape descriptor and SVM is combined in [5] to obtain positive samples. Optical flow is used in [6] for aircraft detection. In [7] local features are used to compute the optical flow. These local features which are updated online are used to make the tracking method robust to target appearance changing. These generative methods are prone to drift in complex scenarios as they do not make fully use of the background information to improve tracking performance.

Some generic tracking methods are tailored for UAV-based tracking. Multiple instance learning is used in [11]. Multiple resolution representation strategy and multiple classifier voting mechanism are employed to handle challenging factors, such as background clutter and illumination changes. Feature point based detection and tracking is proposed in [40, 41]. The TLD tracking is employed in [14]. The experiments show that the background is incorporates into the positive training samples which lead to target drift. Furthermore, the features used in TLD do not consider the aerial tracking. Thus, tracker drifts when scale variation happened and perspective changing. The struck method is employed in [12, 13] and achieves good short term tracking results. However, the experiments are not diverse enough and are specific to certain application domains. In this paper, we will test our tracker on a large dataset that is used for aerial tracking and compare with state-of-the-art tracking methods.

3. Tracking algorithm

The proposed UAV-based tracking algorithm is summarized in Figure 2. The bounding box in the first frame is from the ground truth in the UAV123 dataset. The red bounding box in frame t is the estimated position. The green bounding boxes are the sampling candidates. LARK features are extracted and the two SVM models give the final position via forward and backward tracking. Next the features of training samples are extracted and fed to SVM *model2*.

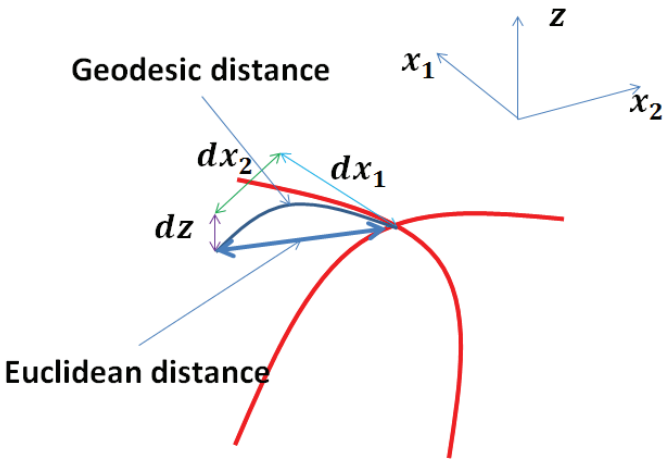

Figure 3. Euclidean distance vs. geodesic distance.

3.1. LARK feature

LARK feature has been used in salience detection [17], object detection [16], face detection [18], action recognition [19]. It can capture the underlying structure of the surrounding data even in the presence of noise and data uncertainty. Different from traditional methods, LARK measures the likelihood of a point to its surroundings based on geodesic distance between points. This distance is the shortest path along a manifold defined by the embedding of the image data in 3D as $\{x_1, x_2, z(x_1, x_2)\}$. Figure 3 shows the differences between geodesic distance and Euclidean distance in the 3-dimensional space. We consider an image surface $S(x_1, x_2) = \{x_1, x_2, z(x_1, x_2)\}$ as shown in Figure 3. 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

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

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

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

where $\Delta x = [dx_1, dx_2]^T$, 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 computation procedure, the pixels center around a local window. The size of the window is small, e.g., 5*5. That means the $\Delta x^T \Delta x$ in the local window is trivial. Thus, $ds^2 \approx \Delta x^T C \Delta x$. The LARK is defined as a self-similarity between a center pixel and its surrounding pixels as follows:

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

where $l \in [1, \cdots, P]$, P is the total number of pixels centered around the central pixel. C is computed by the eigen-

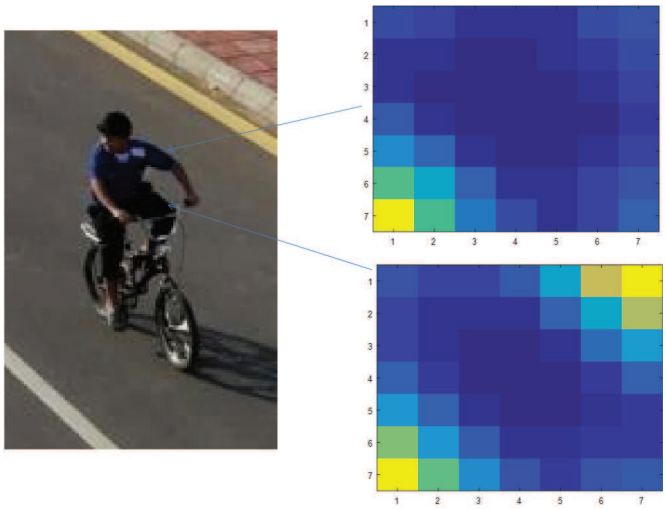

Figure 4. Examples of normalized LARK value.

decomposition as follows,

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

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

LARK computes the oriented gradients which is similar to HOG and SIFT. In HOG and SIFT, quantization is used to oriented gradients, while in LARK, eigen-decomposition enables accurate calculation of the oriented gradients. Meanwhile, the geodesic distance is treated as a similarity measure between the two pixels, which gives a reliable distance metric and explains the promising performance achieved by the previous approaches. Figure 4 demonstrates the LARK kernel. A darker blue color represents a smaller value. The key idea behind the LARK is to robustly obtain local geometric structures by computing the differences based on estimated gradients, which encodes the orientation information of the target. Finally, the LARK kernel computed in each pixel is concatenated to a feature vector to represent the object.

3.2. SVM models

We use the standard SVM with hinge loss and $l2$ regularization [22]. During tracking, two SVM models are used to estimate target position by searching for the maximum classification score around the location from the previous frame. SVM *model1* is only trained in the first frame and does not update during tracking. Our observation is that the bounding box given in the first frame is the target. All the other tracked positions are estimated by ourselves and noise may be introduced. If the model update is contaminated by noise, tracking drift may occur. For example, when a car goes through a tunnel, the appearance will be changed by illumination variation. And the appearance will be recovered once the car goes out of the tunnel. In the object tracking, there are many similar scenarios (e.g., occlusion, pose variation) that the target will be recovered to the initial appearance. Thus, if the SVM model updates too frequently, tracking performance will be degraded. Hence we adopt a conservative model update strategy. Given the estimated position in the current frame, a set of training samples are generated which is used to update the SVM *model2* online.

3.3. Forward and backward tracking

In this section, we give a detailed description of our forward and backward tracking scheme. The key idea of our fusion strategy is to find an objective criterion to measure the accuracy of the trackers. Theoretically, a robust tracking model should be identical to the ground truth during forward and backward tracking. Therefore, forward and backward tracking is adopted in our work to measure the robustness of the two trackers. The backward tracking is initialized at the second frame. By comparing the backward trajectories with the tracked position, we can approximately estimate the accuracy of the forward trackers. Based on the forward and backward analysis, the best forward position is selected. The final results are more robust and accurate than single model tracking.

Figure 5 illustrates two trackers from frame $t-1$ to frame t. We track the target within the red bounding box at frame $t-1$. Black and blue bounding boxes depict the two forward tracking results at frame t. Green and yellow bounding boxes represent corresponding two backward tracking results at frame $t-1$ by SVM *model1* and SVM *model2*. Comparing the overlap between the green bounding box and red bounding box at frame $t-1$, we can tell approximately how much the SVM *model1* matches the object appearance. In the same way, comparing the overlap between the yellow bounding box and red bounding box, we can evaluate the SVM *model2*. The overlaps can be used to evaluate as a fusion weight in the forward tracking. The overlap ratio is computed as follows [45]:

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

where $SVMB_1$ and $SVMB_2$ represent the SVM *model1* and SVM *model2* tracking bounding box in frame t respectively. And PB represents the final tracking bounding box in frame t. $\cup$ and $\cap$ represent intersection and union of the two regions, respectively. In our tracking framework, the final position is determined by the black and blue bounding boxes.

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

where β_1 and β_2 are normalized overlap ratios α_1 and α_2 respectively, y_1 and y_2 are two forward tracking results. The tracking algorithm is given in Figure 6.

Figure 5. The forward and backward tracking.

Input: Current position at frame t. SVM model1 and SVM model2.
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.

Figure 6. Our tracking algorithm.

4. Experiments

4.1. Dataset

Generic tracking datasets such as OTB50 [23], OTB100 [24], TC128 [42], VOT16 [43] and ALOV300 [26] have been widely used to compare with state-of-the-art tracking methods. But the tracking scenarios are different from UAV-based tracking. The VIVID dataset [36] is dedicated to aerial tracking, but it has only 9 sequences. In our experiment, we use two dataset: UAV123 dataset [44] and UAV123-10fps dataset.

The UAV123 dataset contains 123 video data which is recently created for UAV-based target tracking.. The UAV123-10fps is down sampled from the UAV123 dataset. There are 123 videos in this dataset, 115 videos captured by a UAV cameras and 8 sequences rendered by a UAV simulator, which are all annotated with bounding boxes and 12 attributes. The longest sequence contains 3085 frames and the total frames are more than 110k frames. Various scenes exist in the UAV123 dataset, such as roads, buildings, field, beaches and so on. The targets include aerial vehicles, person, trucks, boats, cars and so on. The tracking scenarios contain occlusion, background clutter, camera motion, illumination variation, scale variation, viewpoint change, and so on.

We also compare these trackers in 12 attributes. That is:

Aspect ratio change, which means the fraction of ground truth aspect ratio in the first frame and at least one subsequent frame is outside the range [0.5, 2].

Background Clutter, which is the background near the target has similar appearance as the target.

Camera Motion: abrupt motion of the camera.

Fast Motion: motion of the ground truth bounding box is larger than 20 pixels between two consecutive frames.

Full Occlusion: the target is fully occluded.

Illumination Variation: the illumination of the target changes significantly.

Low Resolution: at least one ground truth bounding box has less than 400 pixels.

Out-of-View: some portion of the target leaves the view.

Partial Occlusion: the target is partially occluded.

Similar Object: there are objects of similar shape or same type near the target.

Scale Variation: the ratio of initial and at least one subsequent bounding box is outside the range [0.5, 2].

Viewpoint Change: viewpoint affects target appearance significantly.

4.2. Compared trackers

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

4.3. Quantitative evaluation

We follow the evaluation strategy in [23]. Two measures are employed to compare the trackers: precision and success. Precision is calculated as the distance between the centers of the ground truth bounding box and a tracker bounding box. Success is computed by the overlap of pixels between the ground truth bounding box and a tracker bounding box. The precision plot represents the percentage tracker bounding boxes within a given threshold distance in pixels of the ground truth. To rank the trackers, we use a threshold of 20 pixels [23]. The success plot represents 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) shows how well the bounding box of a tracker in all frames given the initial bounding box. Overall performance: Figure 7 shows our tracking algorithm achieves 0.644 and 0.402 in precision and success plots of OPE on UAV123 respectively. Figure 8 shows the precision plots of videos with different attributes on UAV123. The number in the title indicates the number of sequences. Figure 9 shows the success plots of videos with different attributes on UAV123. The number in the title indicates the number of sequences. Figure 10 shows On UAV123-10fps, our tracker achieves 0.608 and 0.391 in precision and success plots of OPE respectively. Attribute based performance: Among existing methods, our method performs well with overall success in scale variation (0.435), aspect ratio change (0.347), low resolution (0.289), fast motion (0.28), full occlusion (0.242), partial occlusion (0.342), out-of-view (0.352), background clutter (0.297),

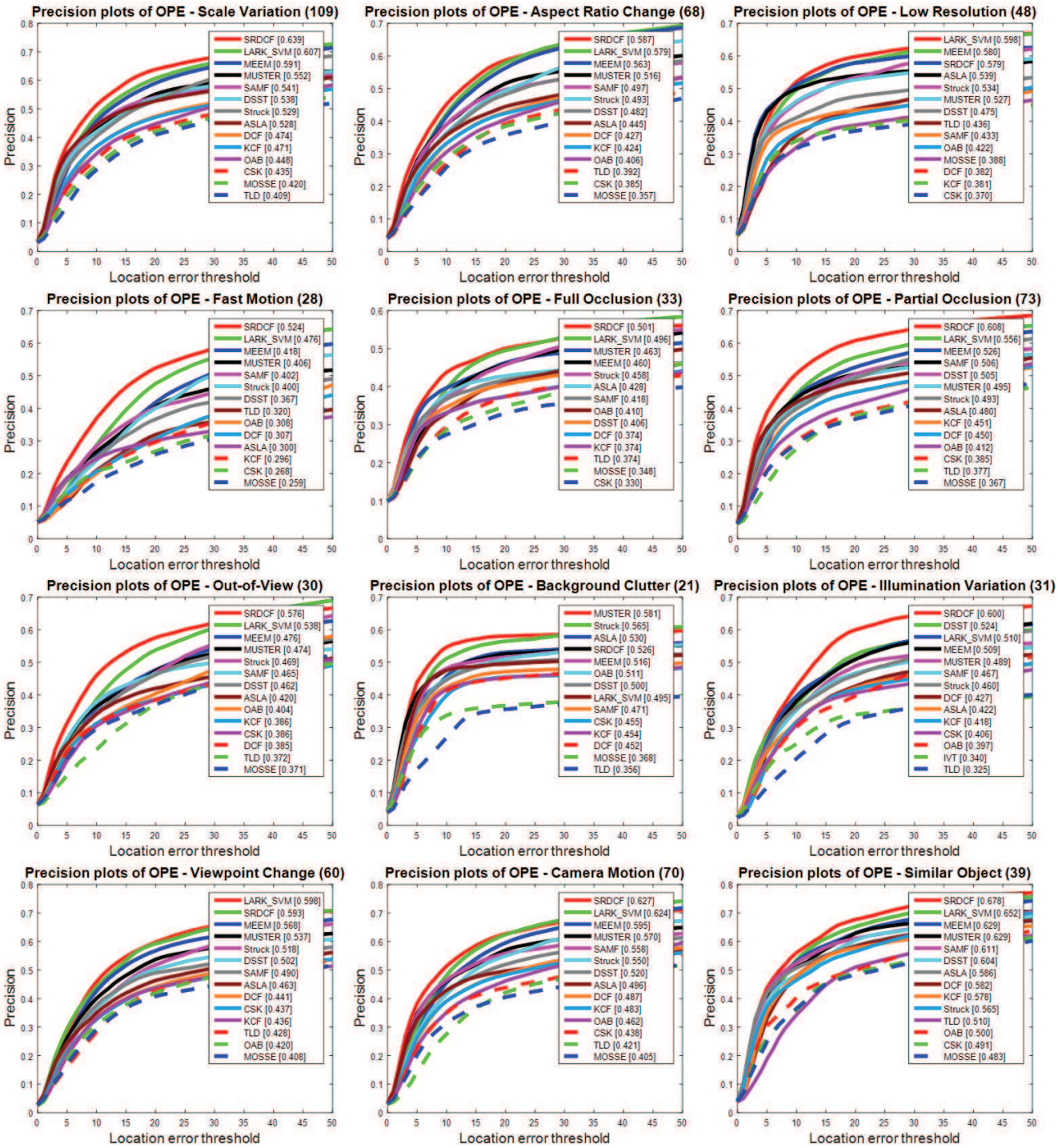

Figure 8. Attribute-based precision plots of OPE on UAV123 dataset. The trackers with best 10 AUC scores are presented in the legends.

illumination variation (0.319), viewpoint change (0.373), camera motion (0.401) and similar object (0.405). In most of the situations, our method yields a favorable tracking performance compared to the other compared trackers.

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 (in Figure 8 and Figure 9). There is still

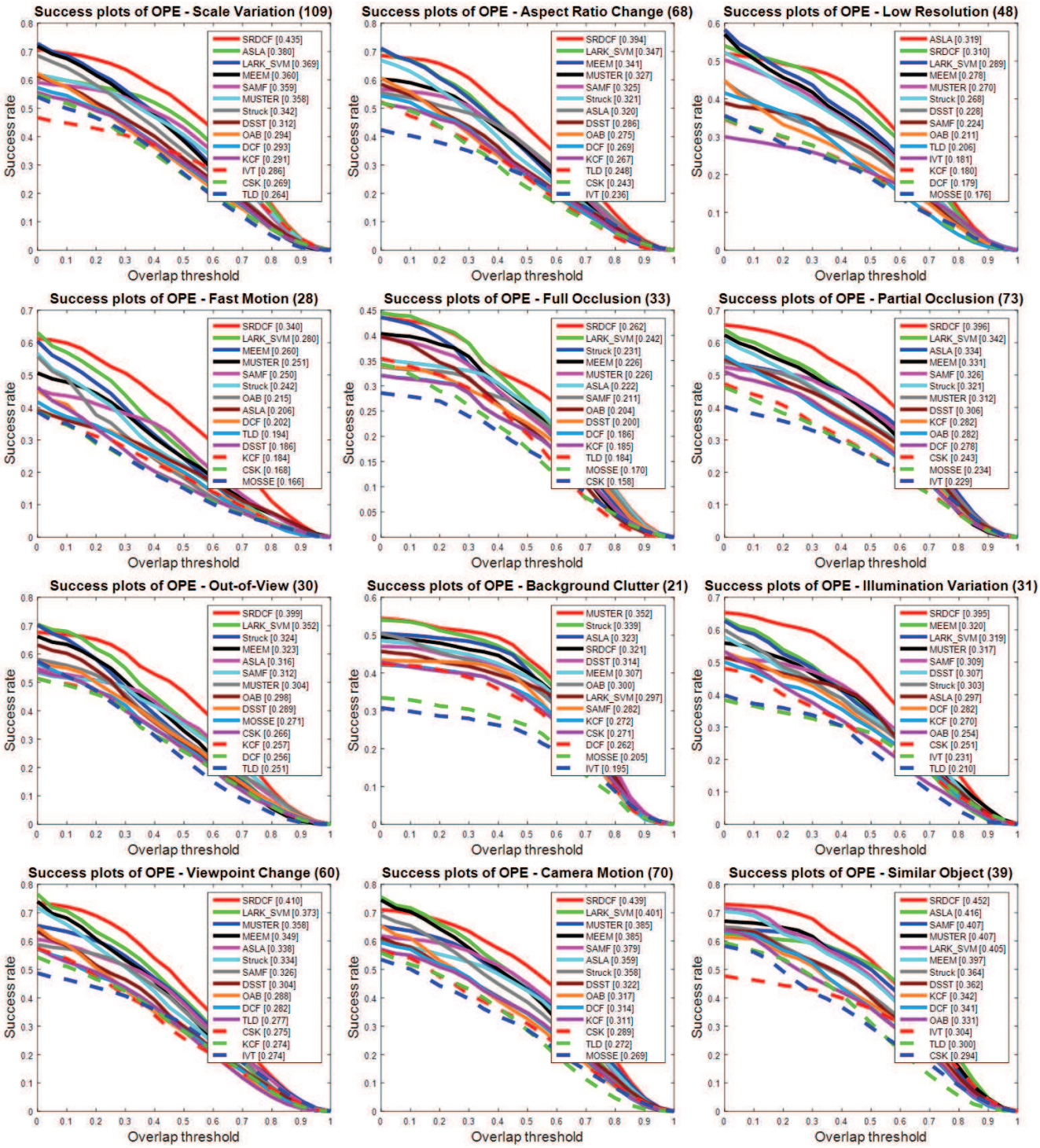

Figure 9. Attribute-based success plots of OPE on UAV123 dataset. The trackers with best 10 AUC scores are presented in the legends.

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.

4.4. Qualitative evaluation

Figure 11 shows sampled results of representative sequences where the targets undergo challenging situations (e.g., pose variation, low resolution, illumination varia-

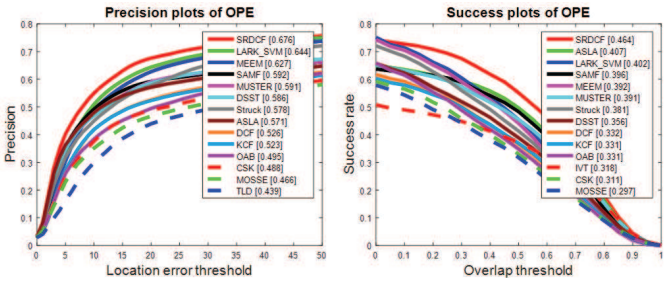

Figure 7. Precision and success plots of OPE on UAV123 dataset.

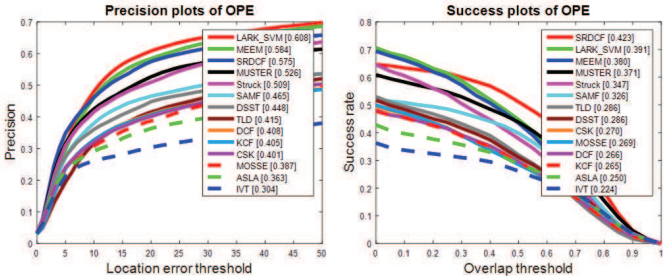

Figure 10. Precision and success plots of OPE on UAV123-10fps dataset.

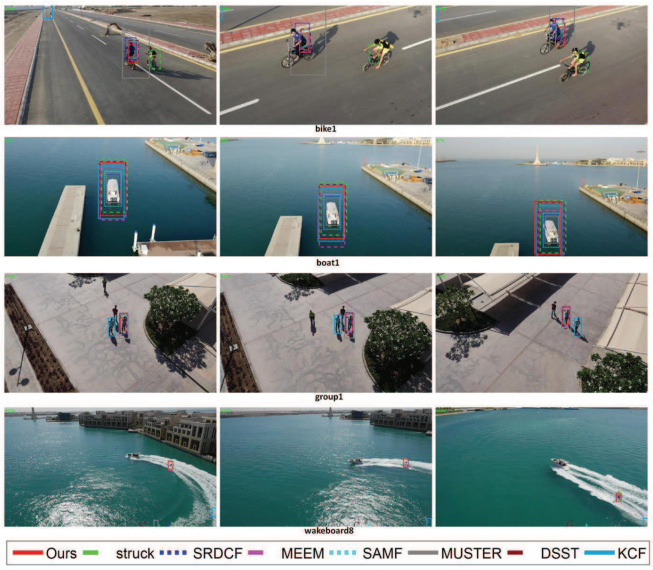

Figure 11. Qualitative results of the 8 trackers over sequences bike1, boat1, group1, wakeboard8, in which the targets undergo severe appearance changes.

tion). In the bike1 sequence, the person undergoes rapid appearance changes due to viewpoint change and illumination variation. Our SVM models are updated online and adapt to the appearance changes. The forward and backward scheme adaptively fuses the tracking results to track the person well. In the boat1 sequence, the target undergoes appearance change due to scale variations. Our method is able to track the boat in the entire video. In the group1 sequence, the person walks while the appearance changes much due to low resolution and illumination variations. Our tracker performs well as the two SVM models adapt to the appearance changes. In wakeboard8 sequences, the target undergoes significant appearance changes due to low resolution and occlusion. The DSST, MUSTER, KCF and SAMF method lock on to the background. The proposed feature captures the edge information of the target and tracks the object well.

5. Conclusion

In this paper, we have introduced a robust UAV-based tracking framework. The discriminative tracking algorithm is achieved by first employing the LARK feature to encode the edge information of the target. Then we use two SVM models to estimate the target position. The forward and backward tracking scheme which is employed to evaluate the tracking accuracy of the two models improve the robustness of the tracking results in various scenarios. The final position is determined by the two models. The positive and negative samples are employed to train and update the SVM *model2* online to adapt to appearance changes. Experiments have shown that the proposed tracking method outperforms most of the existing state-of-art trackers over a range of scenes, lighting conditions and distances in terms of accuracy and robustness.

Acknowledgment

This work was partially supported by a National Institutes of Health (NIH) / National Cancer Institute (NCI) R01 grant (#1R01CA193603). We thank the anonymous reviewers for their careful reading and many insightful comments and suggestions.

References

[1] A. Wainwright and J. Ford. Fusion of morphological images for airborne target detection, in Information Fusion, 15th International Conference on, pp. 1180-1187, 2012.

[2] T. Gandhi, et al. Detection of obstacles in the flight path of an aircraft, IEEE CVPR, vol. 2, pp. 304-311, 2000.

[3] R. Carnie, R. Walker, and P. Corke. Image processing algorithms for uav sense and avoid, ICRA, pp. 2848-2853, 2006.

[4] J. S. Lai, et al. 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] D. Dey, et al. 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] J. W. Mccandless. Detection of aircraft in video sequences using a predictive optical flow algorithm, pp. 523-530, 1999.

[7] A. Mian. Realtime visual tracking of aircrafts, in Digital Image Computing: Techniques and Applications, pp. 351-356, 2008.

[8] I. F. Mondragon, et al. 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] C. Teuliere, L. Eck, and E. Marchand. 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] A. Kendall, N. Salvapantula, and K. Stol. 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., et al. Robust real-time vision-based aircraft tracking from unmanned aerial vehicles, IEEE ICRA, pp. 5441-5446, 2014.

[12] Lim, H., Sinha, S.N. Monocular localization of a moving person onboard a quadrotor mav, IEEE ICRA, pp. 2182-2189, 2015.

[13] Mueller, M., et al. Persistent aerial tracking system for uavs, IEEE IROS, pp. 1562-1569, 2016.

[14] Pestana, J., et al. 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] N. Dalal and B. Triggs. Histogram of oriented gradietns for human detection, IEEE CVPR, pp. 886-893, 2005.

[16] H. Seo and P. Milanfar. Training-free, generic object detection using locally adaptive regression kernels, IEEE TPAMI, 32(9), pp. 1688-1704, 2010.

[17] H. J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance, Journal of Vision, 9(12), pp. 1-27, 2009.

[18] H. J. Seo and P. Milanfar. Face verification using the lark representation, IEEE Transactions on Information Forensics and Security, 6(4), pp. 1275-1286, 2011.

[19] H.J. Seo and P. Milanfar. Action Recognition from One Example, IEEE TPAMI, 33(5), pp. 867-882, May 2011.

[20] Zhang, Jianming, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization, ECCV, pp. 188-203, 2014.

[21] Bolme, D.S., et al. 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, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark, IEEE TPAMI 37(9), pp. 1834-1848, 2015.

[25] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey, ACM Computing Surveys, 38(4), 2006.

[26] A. W. M. Smeulders, et al. 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, R.S., Yang, M.H. Incremental learning for robust visual tracking, IJCV, 77(1), pp. 125-141, 2008.

[29] Hare, Sam, et al. Struck: Structured output tracking with kernels, IEEE TPAMI 38(10), pp. 2096-2109, 2016.

[30] Henriques, J., et al. Exploiting the circulant structure of tracking-by-detection with kernels, ECCV, pp. 702-715, 2012.

[31] Jia, X., Lu, H., Yang, M.H. 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., et al. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking, IEEE CVPR, pp. 749-758, 2015.

[34] M. Danelljan, et al. Accurate scale estimation for robust visual tracking, in BMVC, 2014.

[35] Danelljan, et al. Learning spatially regularized correlation filters for visual tracking, IEEE ICCV, pp. 4310-4318, 2015.

[36] R. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site, IEEE Intl Workshop on PETS, pp. 17-24, 2005.

[37] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration, IEEE ECCVW, pp. 254-265, 2014.

[38] J. van de Weijer, et al. Learning color names for real-world applications, IEEE TIP, 18(7), pp. 1512-1524, 2009.

[39] Henriques, J.F., 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., et al. On-Board Visual Tracking With Unmanned Aircraft System (UAS). Infotech@Aerospace Conferences, AIAA, 2011.

[42] Liang, Pengpeng, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark, IEEE TIP 24(12), pp. 5630-5644, 2015.

[43] Kristan, M., et al. The VOT2013 challenge results, In CVPRW, pp. 98-111, 2013.

[44] Mueller, Matthias, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking, ECCV, pp. 445-461, 2016.

[45] M. Everingham, et al. The PASCAL VOC2010 Results, 2010.