2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

UAV Target Tracking with A Boundary-Decision Network

Ke Song, Wei Zhang*, and Xuewen Rong
School of Control Science and Engineering, Shandong University, China
songke_vsislab@mail.sdu.edu.cn, davidzhangsdu@gmail.com, rongxw@sdu.edu.cn

***Abstract*—The aspect ratio of a target changes frequently during UAV tracking task, which makes the aerial tracking very challenging. Traditional trackers struggle from such problem as they mainly focus on the scale variation issue by maintaining a certain aspect ratio. In this paper, we propose a novel tracker, named boundary-decision network (BDNet), to address the aspect ratio variation in UAV tracking. Unlike previous work, the proposed method aims at operating each boundary separately with a policy network. Given an initial estimate of the bounding box, a sequential actions are generated to tune the four boundaries with an optimization strategy including boundary proposal rejection, offline and online learning. Experimental results on the benchmark aerial dataset prove that the proposed approach outperforms existing trackers and produces significant accuracy gains in dealing with the aspect ratio variation in UAV tracking.**

I. Introduction

Given an initialized state of a target object in a frame of a video, the goal of tracking is to locate a target in the subsequent frames [1]. cooperate with some other computer vision tasks such as terrain perception [2], visual tracking on a unmanned aerial vehicle (UAV) has enabled many new applications such as security, rescue, monitoring and military. Differing from the generic tracking, UAV target tracking is to locate the target from a low-altitude aerial perspective, which poses new challenges to the tracking problem. Especially, the relative distance/movement between drone and target, as well as the shaking of UAV body, are liable to incur significant variation of scale and aspect ratio. The studies in [3] indicate that scale variation and aspect ratio change are the two most difficult attributes the trackers struggle with in UAV tracking.

Recently, there existed some pioneering work which attempted to tackle the severe scale variation issue in UAV target tracking. For example, Li et al. [4] built a scale pyramid that contains a series of boxes with different scale around the target. Wang and Shi [5] employed the forward and backward scheme to prevent the model drift in complex scenarios. Besides, the scale variation is also a common issue in generic tracking, and many attempts have been made. Li and Zhu [6] proposed a tracker with Scale Adaptive with Multiple Features (SAMF) to pursue the target with different scales, and resizes the samples into a fixed size to compare with the learned model at each frame. Danelljan et al. [7] presented a discriminative scale space tracking (DSST) method that learns the separate correlation filters (CFs) for the estimation of translation and scale about the bounding box. Guez et al. [8] employed a Bayesian inference framework with structural constrains to make the tracker adaptive to scale variation. The action-decision network (ADNet) [9] tried to predict the transform actions to adapt to the scale variation by reinforcement learning. However, the above methods were presented to estimate the scale confined to a fixed number of type, e.g. SAMF [6] defined a scaling pool of 7 scale types, while DSST [7] built a scale pyramid with 33 scale types. The scale changes were estimated in ADNet by maintaining the aspect ratio of the tracking target.

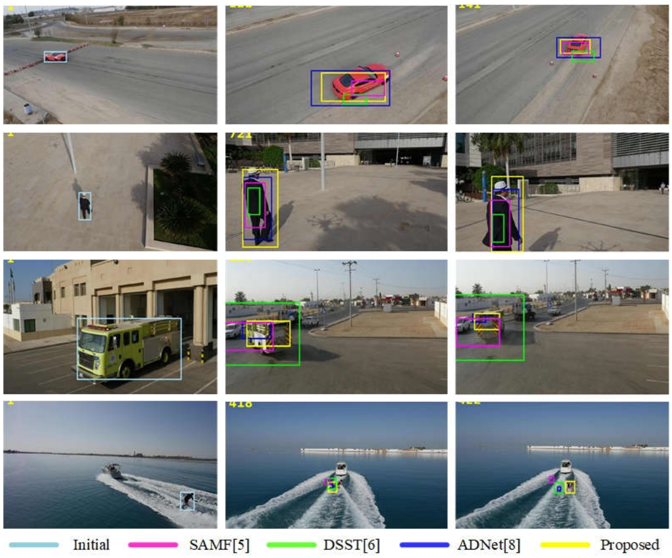

Fig. 1. The video sequences (car18, person20, truck1 and wakeboard5) with the aspect ratio variation issue in the UAV123 dataset. As shown in the first column, the bounding box of each target at the first frame is provided for initialization.

Apparently, most previous trackers mainly focused on the scale variation problem, and the work on aspect ratio change is rare. Whereas, aspect ratio variation is also a common issue that bothers the trackers especially from the aerial perspective. The studies in [3] show that 55% sequences of the benchmark aerial dataset UAV123 [3] have aspect ratio change attribute, i.e., the fraction of ground truth aspect ratio in the first frame and at least one subsequent frame is outside the range of $[0.5, 2]$. For the generic tracking dataset like OTB100 [1], 16% sequences are with aspect ratio change attribute. Hence, the adaptability to aspect ratio variation is crucial for the accuracy

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

of UAV tracking.

Four sequences with aspect ratio variation are selected from UAV123 and shown in Figure 1. It clearly shows that, the scale-adaptive tracker ADNet [9] as well as DSST [7] and SAMF [6] can hardly address the aspect ratio change in UAV tracking. As aforementioned, this is because the aspect ratio is normally defined as constant explicitly or implicitly based on the previous state when the bounding box goes through scale change operation.

In this paper, we propose a novel UAV tracker, named boundary-decision network (BDNet), to address the aspect ratio change in aerial tracking tailored from the generic tracker ADNet. Unlike ADNet that dominates the change of the entire bounding box, the proposed network aims at operating each single boundary separately. The proposed BDNet is designed with a policy network to yield sequential actions to tune all boundaries of the bounding box. We define two types of rewards based on the adjusting result at the terminal step to train the parameters with stochastic gradient ascent. Evaluations are performed on the benchmark UAV tracking dataset UAV123 [3] which provides a comprehensive sampling of tracking nuisances that are ubiquitous in low-altitude aerial scenarios. Extensive quantitative results demonstrate that the proposed network obtains significant gains in dealing with the aspect ratio change compared to ADNet as well as the other state-of-the-art trackers. It also exhibits superiority in dealing with the scale variation and occlusion issues that frequently appear in UAV tracking.

II. Related work

A. Scale-adaptive trackers

The literature for visual object tracking is plentiful, so herein we only focus on the scale adaptive trackers, which are most related to the problem addressed in this work.

SAMF[6] and DSST[7] are two commonly used trackers for scale estimation. SAMF is based on kernelized correlation filters (KCF) [10] and compares the response of 7 candidate patches with different scales for final output. DSST utilizes a separate 1-dimensional scale correlation that computes correlation scores in the scale dimension. These scores are then used to estimate the target scale. Besides, Tang and Feng [11] render the search of correct object scale in a continuous scale space as an optimization problem without building image pyramids in advance. Zhang et al. [12] presented a robust scale estimation method by averaging the scales over consecutive frames. Wang et al. [13] proposed a new adaptive and selective update mechanism to update the translation filters effectively. Besides, similar to [14], part-based trackers [15], [16] also have been proposed. As a summary, the fixity of scale changes restricts the capability of filter-based methods to cope with aspect ratio variation.

B. UAV target trackers

Recently, there existed some work which aimed at the UAV target tracking problem. Some pioneering attempts were made in [17] and [18] based on the tracking-learning-detection (TLD) trackers [19]. Hence, they share the common limitations of TLD and are mainly for the short-term tracking tasks. In contrast, the long-term UAV tracker in [4] explored the connection between the frequency domain tracker and spatial domain detector, and presented a coarse-to-find redetection scheme. Xiao et al. [20] utilized the random forest for features selection, and then transformed the features to binary code for tracking. Wang et al. [5] employed the locally adaptive regression kernel (LARK) feature to encode the edge information of the target, and introduced a forward-and-backward tracking scheme to handle the object deformation. To further improve the accuracy of UAV tracking, the above three methods relied on SVM to distinguish object from background, and in [4] and [20], the scale variation was addressed by scale pyramids similar to the DSST[7].

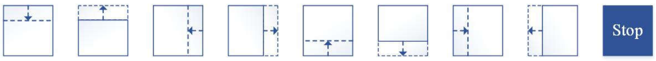

Fig. 2. The nine actions defined to tune the bounding box boundary.

C. ADNet tracker

Deep Learning has been applied with success in computer vision such as object detection [21], person re-identification [22], and visual tracking [9], etc. Our approach is mainly extended from the ADNet [9], which proposed to learn an optimal tracking policy by maximizing the cumulative rewards via reinforcement learning (RL). Starting with the location of the bounding box in previous frame, the policy network yielded an action to transform the bounding box including location and scale in the current frame, which maintains the aspect ratio of the tracking target. Hence, it missed the aspect ratio change problem. The proposed policy network shares the similar spirit with the ADNet by rendering the tracking task into the progress of an agent interacting with the environment. However, as aforementioned, we aim to free the agent and make the bounding box change more flexibly by operating on the boundaries directly with a new policy network.

III. Method

A. Problem setup

As aforementioned, the proposed method aims at relieving the aspect ratio change issue that limits ADNet in UAV tracking tasks. Starting with the output of ADNet as an initial bounding box, the proposed tracker proceeds by generating a policy to move each boundary separately for further improving the tracking result.

As the future mostly depends on the current state and action, such tracking task can be formulated as a Markov Decision Process (MDP) defined by the 5-tuple (S,A,P,R,γ). Similar to [8], S denotes the set of states, A is the set of actions, P denotes the state transition probability kernel, R is a bounded reward function, and γ is the discount factor. The detailed setup are as follows:

State We use l to refer to the l_{th} frame and t to the t_{th}

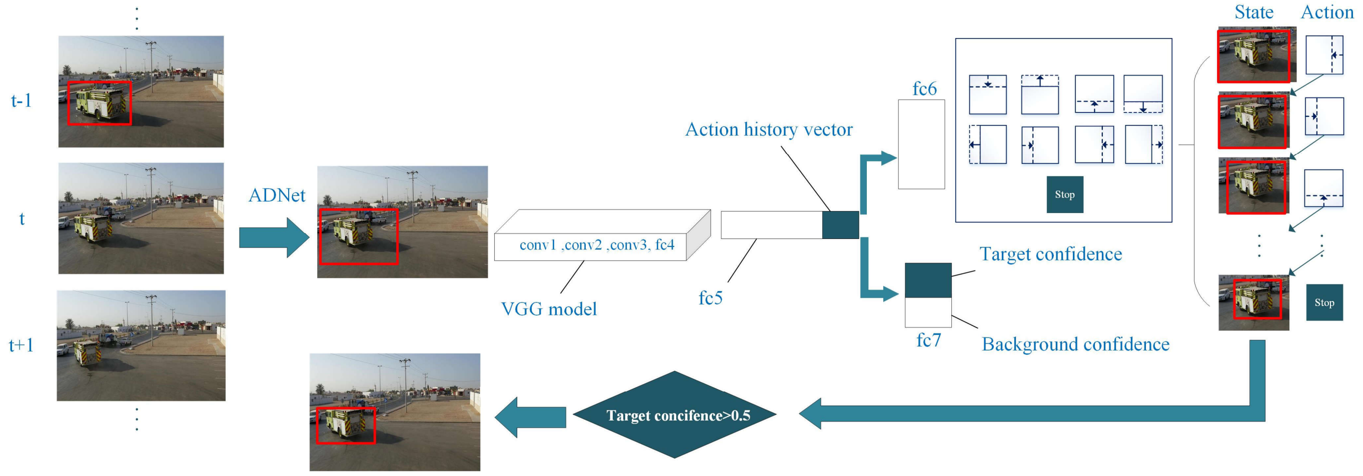

Fig. 3. Illustration of the proposed boundary-decision network for UAV tracking.

iteration in the current frame. The state s_t contains two types of status information: One is the appearance information p_t inside the bounding box in the current iteration (i.e., the patch surrounded by the bounding box). Cropped from the frame, the patch is fixed to 112 × 112 to feed into the network. The other one is the action history information h_t, which stores the last 10 actions accepted during iteration.

Action The agent selects the corresponding action based on the current state. There are totally nine actions as shown in Figure 2. Beside the stop action, each boundary has two movement modes: expand outwards and move inward depending on the relative direction. The length of movement is 0.03 times of the length of corresponding side (height or width). The action vector encode 9 actions, and only the selected action is set to 1 with the others set to 0.

State transition probability We set the state transition probability to 1, which implies the agent should performs the selected action, and each (s_t,a) will dominate the next state s_{t+1}.

Reward The reward is the overall discounted return depending on what the ending state of the agent is. We define the reward r_{s_t} based on the intersection-over-union (IoU), which can well measure the area of intersection between the predicted bounding box and the ground truth.

$$r_T = \begin{cases} +1, & \text{if } IoU(b_T, G) > 0.7 \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

where T is the termination step of iteration, b_T denotes the patch surrounded by the bounding box. G is the ground truth bounding box of current frame.

Factors The discount-factor is maintained zero during the iteration. That is, the the current action has no impact to the future rewards as we only focus on the state at the terminal step.

B. Network architecture

The architecture of our network is illustrated in Figure 3. We employ the pretrained VGG-M model [23] as the initial network, including three convolutional layers {conv1,conv2,conv3} and two fully connected layers {fc4,fc5}. The convolution layers are identical to the corresponding parts of VGG-M network. The next two fully connected layers have 512 output units and are combined with ReLUs and dropouts. Additionally, the fc5 layer concatenates the action history vector to constitute the state. Softmax is employed to form the fc6 layer, which has nine output units corresponding to the actions shown in Figure 2. Given the state, the fc6 layer is able to predict the conditional action probability distribution for all actions. As shown in Figure 3, starting from the initial bounding box, the proposed network proceeds by iteratively tuning the location of box boundaries with a reinforcement learning scheme. The agent selects actions sequentially and updates the tracking states until the stop action is chosen or the oscillation case appears. Given the current state, the confidence layer (fc7) obtains the probabilities of the target and the background. Such probability serves as the confidence score and effects in two ways: 1) determine whether to adopt the tracking adjustment or not; 2) distinguish positive and negative samples for online adaptation of tracking.

C. Training of BDNet

The training of the proposed network consists of two stages: offline training and online training.

Offline training As stated in [24], pre-training a policy networks with supervised learning would provide fast, efficient learning updates with immediate feedback and high-quality gradients. Hence, supervised training is first employed as follows. We collect samples by adding Gaussian noise to the ground truth of each frame. All samples are given two types of labels because the final two parallel layers {fc6,fc7} correspond to two different tasks. To train the fc6 layer, the action that makes the sample obtain maximum IoU is set as label l_{fc6}. A binary label l_{fc7} is employed to train the fc7 layer: l_{fc7} is set to 1 if the sample has an IoU higher than

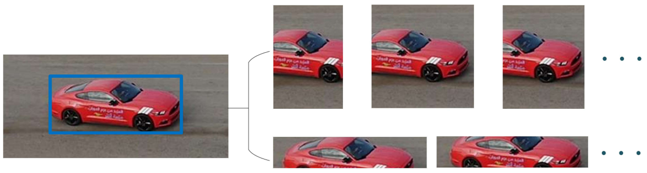

Fig. 4. Manually generate a series of non-optimal tracking samples for training by changing the aspect ratio.

0.7, and otherwise is set to 0. Finally, we train the network parameter W_{SL} by minimizing the multi-loss function L_1 as follows

$$L_1 = \frac{1}{m}\sum_{j=1}^{m} L(l_{fc6}, l_{fc6}^{*}) + \frac{1}{m}\sum_{j=1}^{m} L(l_{fc7}, l_{fc7}^{*}), \qquad (2)$$

where m denotes the batch size, j is the index of samples, L denotes the cross-entropy loss, l_{fc6}^{*} and l_{fc7}^{*} denote the predicted action and class by BDNet respectively. It is noted that the training batch is composed by the frames selected randomly without sequential information, and thus the action history vector in fc5 layer is set to zero herein.

With the pre-trained network by supervised learning, we fix the parameters of the fc7 layer and further tune the proposed network using the policy gradient approach. The training set for policy learning is built in two manners. First, the proposed network aims to reduce the difference between the ground truth and the precomputed bounding box which is obtained by ADNet in the experiments. Hence, we only select the tracking samples whose IoU is higher than 0.5 as the training data. That is, in these frames, the ADNet can mostly capture the target and there is potential to further refine the bounding box through boundary tuning. Second, to enrich the training set, we manually generate a series of non-optimal tracking results whose bounding box differs from the ground truth in aspect ratio as shown in Figure 4. During the policy learning process, the agent can generate a set of sequential states s_t and the corresponding actions a_t for the iteration $t = 1, ..., T$. The action a_t for the state s_t is assigned by

$$a_t = arg\max_{a} p(a|s_t; W_{RL}), \qquad (3)$$

where $p(a|s_t)$ denotes the conditional action probability predicted by the fc6 layer. When the training is done, the final reward r_T is given depends on IoU as shown in Eq. (1), where +1 for success and -1 for failure. The reward will be employed to update the policy network parameters W_{RL} by stochastic gradient ascent as below

$$\Delta W_{RL} \propto \sum_{t}^{T} \frac{\partial \log p(a_t|s_t; W_{RL})}{\partial W_{RL}} r_T. \qquad (4)$$

Online training The network is further updated online to improve the bounding box estimate in a supervised manner similar to the one at the offline stage. Specifically, we fix the parameters of the convolutional filters {w1, w2, w3} and fine-tune the fully-connected layers {f4, ..., f7}.To generate the labeled samples, we utilize the frames whose target confidence score (output of the fc7 layer) is above 0.5, and the current output bounding boxes of these frames are regarded as the ground truth. Then, same to the offline supervised stage, the training set are formed by adding Gaussian noise to the ground truth bounding box of each frame, which has two types of labels: the action label l_{fc6} and the class label l_{fc7}. With these samples and labels, the network is finetued with Eq. (2).

D. Algorithm details

The procedure of the proposed tracker is presented in Algorithm 1. Given a certain frame I_l and the previous tracking result B_{l-1}, the ADNet is first employed to yield a bounding box estimate $B_l^{'}$ which serves as an initial for the next steps. Then, all parameters are set as shown in line 3 to line 6, where d_t denotes the action history vector and ϕ crops the patch surrounded by the initial bounding box $B_l^{'}$ and resizes it to 112 × 112. Next, the policy network updates the action and the state repeatedly as shown in line 8 to line 13. Eventually, when the stop action is chosen or the oscillation state appears, the final result is determined, i.e., either B_l^{*} or $B_l^{'}$ depending on the confidence score yielded by the fc7 layer.

Algorithm 1: Overall tracking procedure

```
   Input:
       The current video frame $I_l$
       Bounding box of the previous frame $B_{l-1}$
   Output:
       Bounding box of the current frame $B_l$
 1 Initial bounding box estimate $B_l^{'}$ with ADNet
 2 Initialize:
 3     $t \leftarrow 1$
 4     $p_t \leftarrow \phi(B_l^{'}, I_l)$
 5     $d_t \leftarrow$ zero setting
 6     $s_t \leftarrow (p_t, d_t)$
 7 repeat
 8     $a_t \leftarrow arg\ max_a\ p\ (a|s_t)$
 9     $b_{t+1} \leftarrow$ Move $b_t$ with $a_t$
10     $p_{t+1} \leftarrow \phi(b_{t+1}, I_l)$
11     $d_{t+1} \leftarrow$ Concatenate $d_t$ with $a_t$
12     $s_{t+1} \leftarrow (p_{t+1}, d_{t+1})$
13     $t \leftarrow t + 1$
14 until stop action or oscillation appears;
15 Output: the updated bounding box estimate$B_l^{*}$
16 if confidence score$\geq$0.5 then
17     $B_l \leftarrow B_l^{*}$
18 else
19     $B_l \leftarrow B_l^{'}$
20 end
```

IV. Experimental Results

A. Dataset and evaluation

The focus of this work is to address the aspect ratio variation issue in aerial video dataset and boost the performance of

UAV tracking. Hence, evaluation was conducted on the UAV tracking dataset UAV123 [3]. We follow the testing strategy in [1] for evaluation and comparison. We compare the proposed method with some state-of-the-art trackers such as: SRDCF [25], ASLA [26], SAMF [6], MEEM [27], MUSTER [28], Struck [29], DSST [7], TLD [19] and ADNet [9]. The experiment is implemented with 2.60GHz Intel Xeon E5-2670 and NVIDIA GeForce GTX 1080 Ti GPU.

B. Analysis on different tracking attributes

There are totally twelve tracking attributes annotated in UAV123, such as scale variation, aspect ratio variation, illumination change, and occlusion. As shown in Figure 4, due to the page limits, we mainly discuss the influence of aspect ratio variation, and the other two most related attributes, i.e., scale variation and occlusion (partial and full), which may also incur aspect ratio change. Besides, these three attributes are the top-3 dominant ones in UAV123.

Aspect ratio change: In UAV123, the aspect ratio change (ARC) attribute is assigned to a frame if the fraction of ground truth aspect ratio in the first frame and at least one subsequent frame is outside the range of $[0.5, 2]$. Hence, as aforementioned, traditional trackers struggle with such attribute, as the aspect ratio is often maintained when refining the bounding box. As shown in the first row of Figure 4, the proposed BDNet produced the best results in terms of both precision and success plots, which demonstrates the superiority over traditional trackers in handling the aspect ratio change issue. Compared to the ADNet, the gains achieve 3.9% and 4.0% in precision and success plots, respectively. This is because the proposed policy network can tune each boundary of the bounding box freely and flexibly, and thus is less prone to the aspect ratio change issue.

Scale variation: As shown in the second row of Figure 4, the proposed BDNet also produced the best results in dealing with the scale variation (SV). Compared to the ADNet, the gains are 3.2% and 3.4% in precision and success plots, respectively. This is a byproduct of our proposed boundary-decision network. As the boundary could be tuned separately, it could become a remedy to the inaccurate scale estimate of the bounding box. Hence, the scale variation issue could be relieved as well.

Partial occlusion and full occlusion: The partial occlusion (POC) and full occlusion (FOC) may also lead to aspect ratio variation of target. Our method still outperforms the other trackers except that SRDCF performed slightly higher precision plot than ours in the partial occlusion attribute. Compared to the above two attributes, the gains over ADNet is less impressive. This is because the occlusion can be considered as an abrupt deformation of the target, which increases the difficulty of improving the tracking accuracy by boundary tuning only.

C. Overall evaluation

The overall performance of the proposed network on the whole dataset with all twelve attributes is evaluated in Figure 6. Again, our method performs favorably against the other state-of-the-art trackers in terms of precision and success plots. Besides, we also make comparison with the LARK_SVM [5], a tracker for UAV aerial scenarios published in ICCV17. Since its codes are unavailable, we only compare the overall performance reported in the paper as shown in Table I without plots. Apparently, our proposed tracker works better than the LARK_SVM [5] overall and in all attributes.

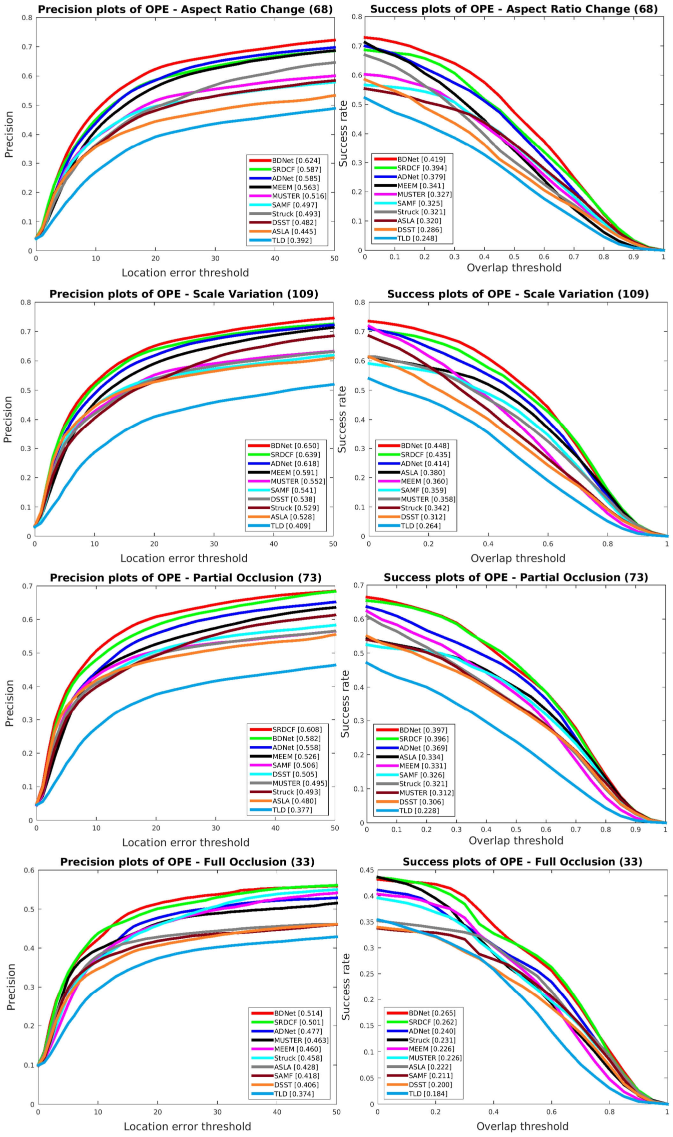

Fig. 5. Performance evaluation of existing trackers with different attributes of UAV123 in terms of precision and success plots.

For the computational efficiency, the whole tracker (ADNet+BDNet) runs at the speed of 2.3 fps on average in UAV123, which is slightly slower than using the ADNet only which runs at 3 fps. Hence, a small sacrifice in efficiency could boost the tracker in a considerable way. Please visit http://www.vsislab.com/projects/robot/UAVTracking/ for videos of the UAV tracking results.

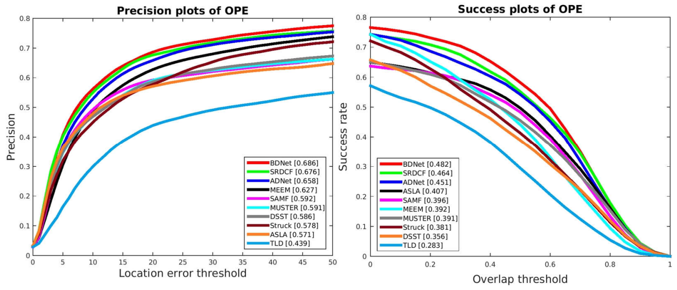

Fig. 6. Overall evaluation of existing trackers in UAV123.

TABLE I
COMPARISON WITH THE BENCHMARK UAV TRACKER LARK_SVM [5].

	ARC	SV	POC	FOC	Overall
Proposed	0.419	0.448	0.397	0.265	0.482
LARK_SVM [5]	0.347	0.369	0.342	0.242	0.402

V. CONCLUSIONS

In this paper, we presented a new tracker to address the aspect ratio variation issue for the target tracking in UAV. Instead of dominating the entire bounding box, the proposed network focused on generating a policy to move each boundary flexibly to adapt to the target with varied aspect ratio. Experiments were conducted on UAV123, which is a benchmark aerial dataset specializing in the low-altitude visual tracking problem. The results show that our approach boosted the tracking performance in terms of accuracy in dealing with the aspect ratio change as well as the scale variation and occlusion.

ACKNOWLEDGMENT

This work was supported by the NSFC Grant No.61573222, Shenzhen Future Industry Special Fund JCYJ20160331174228600, Major Research Program of Shandong Province 2015ZDXX0801A02, National Key Research and Development Plan of China under Grant 2017YFB1300205 and Fundamental Research Funds of Shandong University 2016JC014.

REFERENCES

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[2] W. Zhang, Q. Chen, W. Zhang, and X. He, "Long-range terrain perception using convolutional neural networks," *Neurocomputing*, vol. 275, pp. 781–787, 2018.

[3] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 445–461.

[4] R. Li, M. Pang, C. Zhao, G. Zhou, and L. Fang, "Monocular long-term target following on uavs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 29–37.

[5] Y. Wang, W. Shi, and S. Wu, "Robust uav-based tracking using hybrid classifiers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2129–2137.

[6] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration." in *ECCV Workshops (2)*, 2014, pp. 254–265.

[7] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[8] A. Guez, D. Silver, and P. Dayan, "Efficient bayes-adaptive reinforcement learning using sample-based search," in *Advances in Neural Information Processing Systems*, 2012, pp. 1025–1033.

[9] S. Y. J. C. Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," 2017.

[10] 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.

[11] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3038–3046.

[12] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 127–141.

[13] X. Wang, Z. Hou, W. Yu, Z. Jin, Y. Zha, and X. Qin, "Online scale adaptive visual tracking based on multilayer convolutional features," *IEEE Transactions on Cybernetics*, 2017.

[14] W. Zhang, B. Ma, K. Liu, and R. Huang, "Video-based pedestrian re-identification by adaptive spatio-temporal appearance model," *IEEE transactions on image processing*, vol. 26, no. 4, pp. 2042–2054, 2017.

[15] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4312–4320.

[16] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912.

[17] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision based gps-denied object tracking and following for unmanned aerial vehicles," in *Safety, security, and rescue robotics (ssrr), 2013 ieee international symposium on*. IEEE, 2013, pp. 1–6.

[18] K. Haag, S. Dotenco, and F. Gallwitz, "Correlation filter based visual trackers for person pursuit using a low-cost quadrotor," in *Innovations for Community Services (I4CS), 2015 15th International Conference on*. IEEE, 2015, pp. 1–8.

[19] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[20] Q. Xiao, Q. Zhang, X. Wu, X. Han, and R. Li, "Learning binary code features for uav target tracking," in *Control Science and Systems Engineering (ICCSSE), 2017 3rd IEEE International Conference on*. IEEE, 2017, pp. 65–68.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[22] W. Zhang, X. Yu, and X. He, "Learning bidirectional temporal cues for video-based person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, doi: 10.1109/TCSVT.2017.2718188.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[25] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[26] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1822–1829.

[27] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

[28] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758.

[29] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.