A Real-time Robust Approach for Tracking UAVs in Infrared Videos

Han Wu Weiqiang Li Wanqi Li Guizhong Liu
SigPro Lab, School of Information and Communication Engineering, Xi'an Jiaotong University
Xianning West Road 28, 710049, Xi'an, P.R. China
xjtuwh@stu.xjtu.edu.cn lwq1230@stu.xjtu.edu.cn wanqili@stu.xjtu.edu.cn liugz@xjtu.edu.cn

Abstract

Object tracking has been studied for decades, but most of the existing works are focused on the RGB tracking. For an infrared video, the object is often textureless, especially for far-range drone planar targets. Furthermore, motion of camera and unexpected movement of the drones make tracking more difficult, causing existing object tracking algorithms lose the targets. In this paper a robust and real-time tracking algorithm is proposed for infrared drones, in which a feature attention module and an expansion strategy for searching the target are added to the fully convolutional classifier. Experiments on the Anti-UAV infrared dataset show its robustness to the different challenges of real infrared scenes with a high efficiency.

1. Introduction

Recently, the amount of aunmanned aerial vehicles (UAVs) is growing rapidly because of their autonomy, flexibility, and a broad range of applications. Camera-equipped drones are rapidly deployed in various applications, including smart agriculture, aerial photography, person search and rescue, reconnaissance and surveillance. Computer vision on UAVs has received more and more attention due to their wide applications. At the same time, we also need to be aware of the potential threat to our lives caused by UAVs intrusion. The development of drone technology will inevitably promote the development of counter-drone technology, and the detection and tracking of the drones are becoming increasingly important.

As a basic problem of computer vision, object tracking has received widespread attention. For single object tracking, it mainly focuses on general objects, also known as model-free tracking. In this paper, the tracked objects are the drone. The input is a video sequence and the initial area specified in the first frame, and the tracker determines the position and size of the drone in the subsequent video frames. The tracker should track the drones robustly in real time.

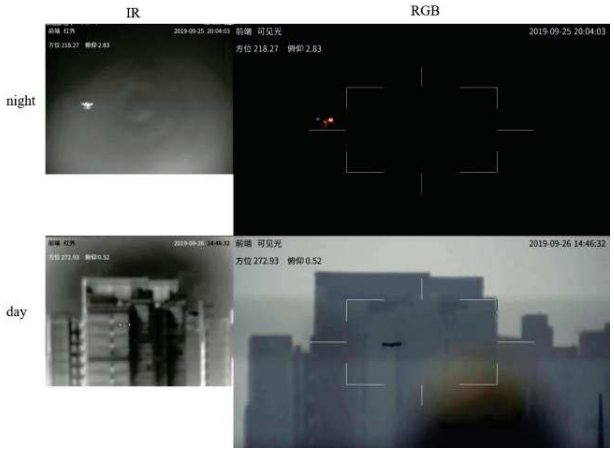

Figure 1. Comparison of infrared and color images. On the left are infrared frames. The resolution is 512 x 640. On the right are the color images. The resolution is 1080 x 1920 (proportional scaled down for display). The images on first row are taken at the same moment in the night. The images on second row are taken at the same moment in the day. From top to bottom: Anti-UAV dataset 20190925_200320_1_2, 20190926_144550_1_2.

Some computer vision researches about UAVs utilize the video captured by drones and the objects are persons, cars and so on [1]. The Anti-UAV workshop (https://anti-uav.github.io/) presents a benchmark dataset and evaluation methodology for detecting and tracking UAVs. In this paper, our goal is to track the drone in a video. Drones vary widely in size. A drone near to the camera has a larger imaging area, both the contour profile and the texture are distinct. However, a drone far from the camera has a smaller imaging area, showing the appearance of blobs, with less texture information. The buildings, trees, etc. in the real scene can clutter the background. There are also false similar targets disturbing tracking. Distractors in the video can occlude the target. With the target movement, the size in planar is changing. Sudden movement of the imaging device can cause a large displacement of the target position, even out of view.

Despite the significant application potential, tracking with infrared has received significantly less attention than RGB tracking. The earliest thermal and infrared (TIR) tracking comparisons were organized by the Performance

Evaluation of Tracking and Surveillance (PETS) [3] in 2005. The VOT2015 and VOT2016 introduced the VOT-TIR challenge that focused on short-term infrared tracking [4]. Color cameras and infrared cameras collect different signals, the former imaging is based on the reflected light, and the infrared camera imaging uses the infrared radiation (0.75-13μm) emitted by objects with a temperature above absolute zero [6]. Infrared cameras can penetrate haze, clouds, rain, and fog for imaging, even in the dark, which significantly affect the imaging quality of visible spectrum. Thermal sensors are more effective in capturing objects than visible spectrum cameras under poor lighting conditions and bad weathers, but color images are able to provide richer color and texture information. In order to join the RGB and thermal images for object tracking, the VOT committee bases the VOT-RGBT-challenge on the existing RGBT-dataset published by [6]. The infrared and RGB images in these datasets have been aligned, and the resolution and field of view of them are the same. However, in reality, the positions and viewing angles of the infrared sensors and color sensors are different. Moreover, the resolution of the color images may be different from that of the infrared images due to different sensors. Therefore, the position and size of the target in the RGB and infrared images can be different. Figure 1 shows some typical scenarios.

In this paper we propose an infrared tracking approach, which can track UAVs in complex real scenes robustly. The proposed search strategy can capture the target accurately even if the camera or target suddenly moves fast, causing a large displacement of the target in the frame.

The rest of the paper is organized as follows. We first review the related works in Section 2, and present our approach to infrared UAVs tracking in Section3. Section4 demonstrates the experimental results. We use the dataset from the Anti-UAV workshop in our experiments. Finally, we summarize our work in Section 5.

2. Related works

The research of single object tracking problem has been a long time. Many related works have been thoroughly proposed. We introduce the related works of RGB tracking and infrared tracking in this section.

2.1. RGB tracking

Correlation filter based Methods. The Discriminative Correlation Filter (DCF) based trackers are able to efficiently utilize limited data by including all shifts of local training samples in the learning. DCF-based methods train a least-squares regressor to predict the target confidence scores by utilizing the properties of circular correlation and the Fast Fourier Transform (FFT) at both learning and detection steps [7]. The seminal work that put forward the correlation filter to tracking is MOSSE, which uses a set of samples random affine transformed from the single initial frame to construct a minimum output sum of squared error filter [8]. By diagonalizing the circulant data matrix with the Discrete Fourier Transform, KCF [9] reduces both the storage and computation by several orders of magnitude. The periodic assumption of KCF [9] also introduces unwanted boundary effects, which severely degrades the quality of the tracking model. To mitigate the boundary effects, SRDCF [10] introduced a spatial regularization component in the learning to penalize correlation filter coefficients depending on their spatial location. In addition, there are several other excellent correlation filter based trackers, such as STRCF [11] and ECO [12]. These algorithms usually divide tracking into two stages, the feature extraction and target classification, and they cannot be trained end-to-end. Same as the discriminative correlation filter approaches, the objective function of target classification module in ATOM [13] is mean square error based but it is modeled on a 2-layer fully convolutional neural network. ATOM regresses the size of the target with IoU-Net [14]. Although ATOM has achieved effective performance, it is sometimes wrong for the size estimation because the predicted Intersection over Union (IoU) may be inaccurate, which leads to tracking failure especially in presence of clutter background.

Siamese network based Methods. Recently, trackers based on Siamese network [15] have received much attention due to their satisfactory balance between performance and efficiency for visual tracking. Siamese network learns a similarity metric function from image pairs offline and converts the tracking task to a template matching task. SiamFC [15] utilizes a large number of template and search region paired samples in offline training. In online tracking, through the forward propagation, the template and search region correlate with each other in the feature space and the target location is determined according to the peak position of the cross-correlation response. SiamRPN [16] adds the Region Proposal Network [18] to acquire various aspect ratio candidate target box. It interprets the template branch in Siamese subnetwork as training parameters to predict the kernel of the local detection task, which treats the tracking task as a one-shot local detection task. On the basis of SiamFC and SiamRPN, SiamMask [17] adds a segmentation branch. The size and shape of the target are obtained according to the mask at the position with the largest classification score, and the tracking results are refined. One shortcoming of Siamese approaches is that they ignore the context information around the template; the template information is simply extracted from the initial target area. Due to the unconstrained video conditions like illumination changes, viewpoint changes, the appearance of the subsequent targets may change a lot compared to the initial one. Consequently, previous proposed Siamese

based trackers degrade in presence of similar distractors and object appearance variation. In order to overcome the shortcomings of the Siamese approaches, DiMP [19] trains a discriminative classifier in an online manner to separate the object from the background; the model is derived from a discriminative learning loss by designing a dedicated optimization process, that is capable of predicting a powerful model in only a few iterations. It keeps collecting positive and negative samples along the tracking process whenever the target is predicted with sufficient confidence, and the tracker is updating online every 20 frames or a distractor peak is detected, thereby enabling the tracker to deal with appearance changes effectively.

2.2. Infrared tracking

Frankly, detection and tracking in thermal infrared imagery can be regarded as detection and tracking in grayscale visual imagery. However, the characteristics of thermal infrared radiation and imagery pose certain challenges to image analysis algorithms. Work [20] introduces a template-based tracking method designed specifically for thermal infrared imagery. It uses Generative Adversarial Network (GAN) to convert the labeled RGB images into pseudo-infrared images for data augmentation. They compute motion features as an extra feature channel by thresholding the absolute pixel-wise difference between the current and the previous frame. A local and global attention mechanism is proposed to integrate RGB and infrared images for tracking in [22]. Paper [23] proposes an RGBT (RGB and Thermal) tracking framework based on a deep adaptive fusion network. The proposed recursive fusion chains can adaptively combine features of all the layers in an end-to-end manner. Work [6] proposes a graph-based approach to learn an object representation for RGBT tracking. The tracked object is represented as a graph with the image patches as nodes. Paper [24] proposes an end-to-end tracking framework for fusing the RGB and IR modalities in RGBT tracking. They consider several fusion mechanisms acting at different levels of the framework, including pixel-level, feature-level and response-level. We mainly focus on infrared tracking in this paper.

3. Proposed Infrared Tracking Approach

For a robust and real-time tracker on infrared drones, we propose a fully convolutional classifier for distinguishing the target from the background. A feature attention module and an expansion strategy for searching the drones are added to improve the robustness. A formal description of the whole algorithm is presented as follows.

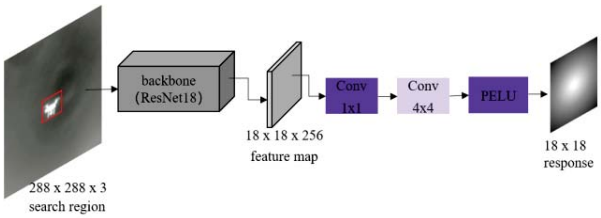

Figure 2. Classifier network architecture

Formal description of the algorithm

i. Input the first frame of a sequence and train the classifier online by the object initial state.
ii. Start tracking in subsequent frames. The backbone network with the feature attention mechanism is used to extract features of the search area. The response map is obtained by the classifier.
iii. Judge tracking success or failure by response peak score.
iv. If tracking fails, start expansion search.
v. Output estimated target state and update the classifier.

The details are given in the subsequent subsections. Subsection 3.1 is the classifier of the proposed tracking approach which discriminates the drones from the background in infrared images. The feature attention mechanism is described in Subsection 3.2. And the expansion searching strategy for target is explained in Subsection 3.3.

3.1. Classifier

The classifier is exclusively trained to discriminate the target from the scene by predicting a target confidence score, based on the backbone features extracted from the current tracking frame. The classifier is composed of a pre-trained feature extraction backbone network and two convolutional layers trained online, shown as in Figure 2. In order to run at real time, the backbone network employs ResNet18 [25] pretrained on ImageNet [26] and the block 4 feature is used for classification. The first convolutional layer in our classifier consists of a 1×1 convolutional layer w_1, which reduces the feature dimensionality to 64. The second convolutional layer employs a 4×4 kernel w_2 with a single output channel. We use a continuously differentiable parametric exponential linear unit (PELU) [27] as output activation: $s = \begin{cases} t, & t \geq 0 \\ \alpha(e^{\frac{t}{\alpha}} - 1), & t \leq 0 \end{cases}$. Setting $\alpha = 0.05$ allows us to ignore easy negative examples in the loss (1).

Same as ATOM [13], the learning objective of the classifier is l^2 classification error based,

$$L(w) = \sum_{j=1}^{N} \gamma_j \| f(x_j; w) - y_j \|^2 + \sum_k \lambda_k \| w_k \|^2 . \quad (1)$$

Each training sample feature map x_j is annotated by the classification confidences $y_j \in \mathbb{R}^{W \times H}$, set to a sampled Gaussian function centered at the target location. Here N is the number of samples. The impact of each training sample is controlled by the weight γ_j, while the amount of regularization on w_k is set by λ_k ,where $\lambda_1 = 0.1$, $\lambda_2 = 0.0001$. The initial weight sum is 1 and the minimum weight sum is 0.25. The regression problem is solved by optimizing a one-channel-output convolution layer [28]. The training samples are sampled from the initial frame with groundtruth and the history frames with the tracked object. Features are always extracted from patches of size 288 × 288 scaled from image regions corresponding to 5 times the estimated target size. To further make our classifier robust in the presence of distractors, we adopt a hard negative mining strategy, common in many visual trackers [29]. If a distractor peak is detected in the classification scores, we double the weights of this training sample and instantly run a round of optimization with standard settings. The object location is then determined according to the peak position of response map.

3.2. Feature attention

With the development of deep learning, hand-crafted feature extraction descriptors are replaced by deep convolutional neural networks. The powerful representation ability of backbone network is essential for all computer vision tasks including object tracking. The backbone network extracts the target structural and the semantic information from the original image space. In this paper, we propose a feature attention mechanism to improve the power of feature representation for tracking robustly. The attention weights for re-calibrating a feature map are directly deployed to the basic block such as the residual module of ResNet18 [25], as shown in Figure 3. Consider a basic block of ResNet18, and denote by X a feature map with axis in the convention order of (C, H, W) (i.e., channel, height and width). The basic block includes a residual module and a skip connection. The residual module consists of 3 × 3 convolutions, BN [31], ReLU [32], 3 × 3 convolutions, and BN [31], which are connected in sequence. Inspired by SENet [33] and CBAM [34], we use the global average pooling (GAP) and the global max pooling (GMP) to generate channel-wise statistics respectively. Both descriptors are then forwarded to a shared network to produce channel attention maps separately. The shared network consists of two fully connected layers (FC) and an ReLU [32] layer. The first FC layer reduces the number of channels and the second FC layer recovers the number of channels to learn a non-mutually-exclusive relationship among the channels.

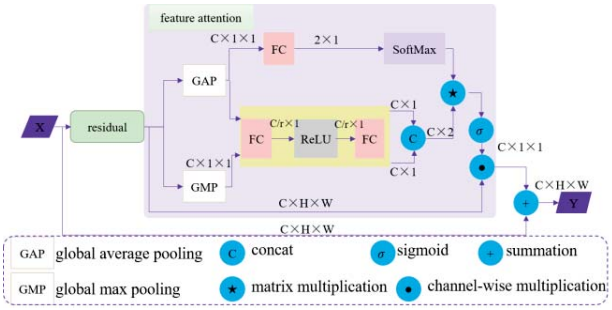

Figure 3. We design a feature attention mechanism for learning instance-specific channel-wise attention weights to re-calibrate the input feature map obtained by residual module.

The ReLU [32] layer learns a nonlinear interaction between channels. Next, the two feature maps are concatenated together along the channel axis, represented by a C × 2 tensor. Since average pooling can use all the information of the feature map adaptively, we use the descriptor of GAP to generate fusion coefficients. In order to fuse the two descriptors adaptively, the descriptor of GAP is forwarded to a FC layer to output the coefficients of linear combination after softmax function. After matrix multiplication, the final weights of the channels are obtained with a sigmoid activation. The output of the feature attention module is obtained by rescaling the residual feature maps with the final weights. And the output Y of the basic block is the sum of X and the output of the feature attention module. Through the attention mechanism, the feature space can be modified adaptively to adopt new appearance features obtained in the course of infrared tracking without overfitting.

3.3. Strategy for searching target

Normally, according to inter-frame smoothness and correlation, the search area of the current frame is centered at the estimated target position in the previous frame, with 5 times the estimated target size. However, in the real scene, the cameras may move suddenly and the drones in the field of view may also move quickly, causing a large scale displacement in any direction between two adjacent frames which break the smoothness of the trajectory of drones. Once it happens, the above search area may not contain the object and the tracking confidence will be low. If the tracking confidence falls below 0.1, we will change the center of the search region. Shown as Figure 4, the search center is shifted left, right, up, and down relative to the initial search center tentatively, and the step size of the shifts is 2.5 times the maximum displacement of the target between two adjacent historical frames. To avoid introducing distractors, the search area is still 5 times the estimated target size.

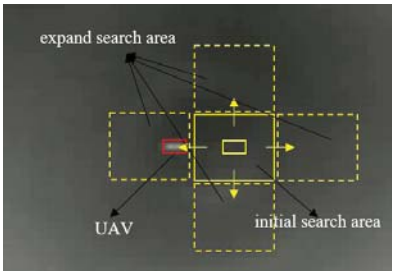

Figure 4. The yellow solid line box indicates the initial search area in the current frame. But the UAV indicated by the red box is not in the initial search area. The initial search center is shifted left, right, up, and down relative to the initial search center tentatively, represented by the four yellow dotted boxes, to find the UAV.

The response peaks of the four additional search areas are compared to get the maximum value. We use the peak value obtained in the second frame as a reference value. If the maximum value is greater than 0.8 times the reference value, the offset distance corresponding to the peak is regarded as the object offset from the search center. Otherwise, we change the center of the search area again, and the step size of the shifts is 5 times the maximum displacement of the target between two adjacent historical frames. If the maximum value is greater than 0.8, the offset corresponding to the peak is regarded as the target offset from the new search center, otherwise, the target location is still the initial search result conservatively. With the expansion search strategy, our algorithm not only performs tracking stably, but also can solve the problem of large-scale displacement of the target due to sudden motion of camera or target.

4. Experiments

We perform a lot of experiments on Anti-UAV infrared dataset and evaluate the performance of the proposed tracking approach.

4.1. Implementation details

The backbone network ResNet18 [25] with proposed attention mechanism is pre-trained on ImageNet classification dataset [32]. In the first frame, we perform data augmentation by applying varying degrees of translation, rotation, blur, gridMask [35], and dropout, resulting in 31 initial training samples [36]. We then apply Gauss-Newton and conjugate gradient method [28] to optimize the parameters w . Subsequently, we only optimize the final convolutional layer w_2 every 20th frame. In every frame, we add the extracted feature map x_j as a training sample, annotated by a Gaussian y_j centered at the estimated target location.

Tracker	acc
SiamFC [15]	0.420
ECO [12]	0.518
ATOM [13]	0.572
DiMP18 [19]	0.585
SiamRPN++ [37]	0.651
ours	**0.682**

Table 1. Performance evaluation for algorithms on the Anti-UAV IR dataset. The best result is marked in bold.

Tracker	acc	speed (fps)
ours	0.682	55.6
ours w/o attention	0.669	66.7
ours w/o expand	0.611	62.5

Table 2. Ablation study for algorithms on the Anti-UAV IR dataset.

The weights γ_j in (1) are updated with a learning rate of 0.01. Our proposed tracking algorithm is implemented in PyTorch with 3.50 GHz Intel Xeon(R) CPU E5-1620 and an NVIDIA GTX1080Ti GPU.

4.2. Anti-UAV infrared dataset

The test-dev infrared dataset consists of 100 thermal infrared video sequences, spanning multiple occurrences of multi-scale UAVs. We use the data to evaluate our algorithm. The tracking average accuracy score (acc) is utilized for evaluation. The acc is defined as

$$acc = \frac{1}{T}\sum_{t}(IoU_t * \delta(v_t > 0) + p_t(1-\delta(v_t > 0))). \quad (2)$$

At frame t , the IoU_t is the IoU between the corresponding ground truth and tracking boxes. The v_t is the visibility flag of the ground truth. If the target exists in the current frame, $\delta(v_t > 0) = 1$. When the target does not exist in the frame and $1-\delta(v_t > 0) = 1$, if the tracker's prediction is empty, $p_t(1-\delta(v_t > 0)) = 1$, otherwise, $p_t(1-\delta(v_t > 0)) = 0$. The accuracy is averaged over all the T frames. Our acc score is calculated according to the average results on the 100 IR videos. From Table 1 it is seen that our approach gets the highest average accuracy. The code of SiamFC [15] is the Pytorch version and the model is provided by the Anti-UAV organizer. For ECO [12], ATOM [13] and DiMP18 [19] we use the codes and models released at https://github.com/visionml/pytracking. Their backbone networks are ResNet18 [25]. For SiamRPN++ [37] we uses the codes and models released at https://github.com/STVIR/pysot. And its backbone network is ResNet50 [25].

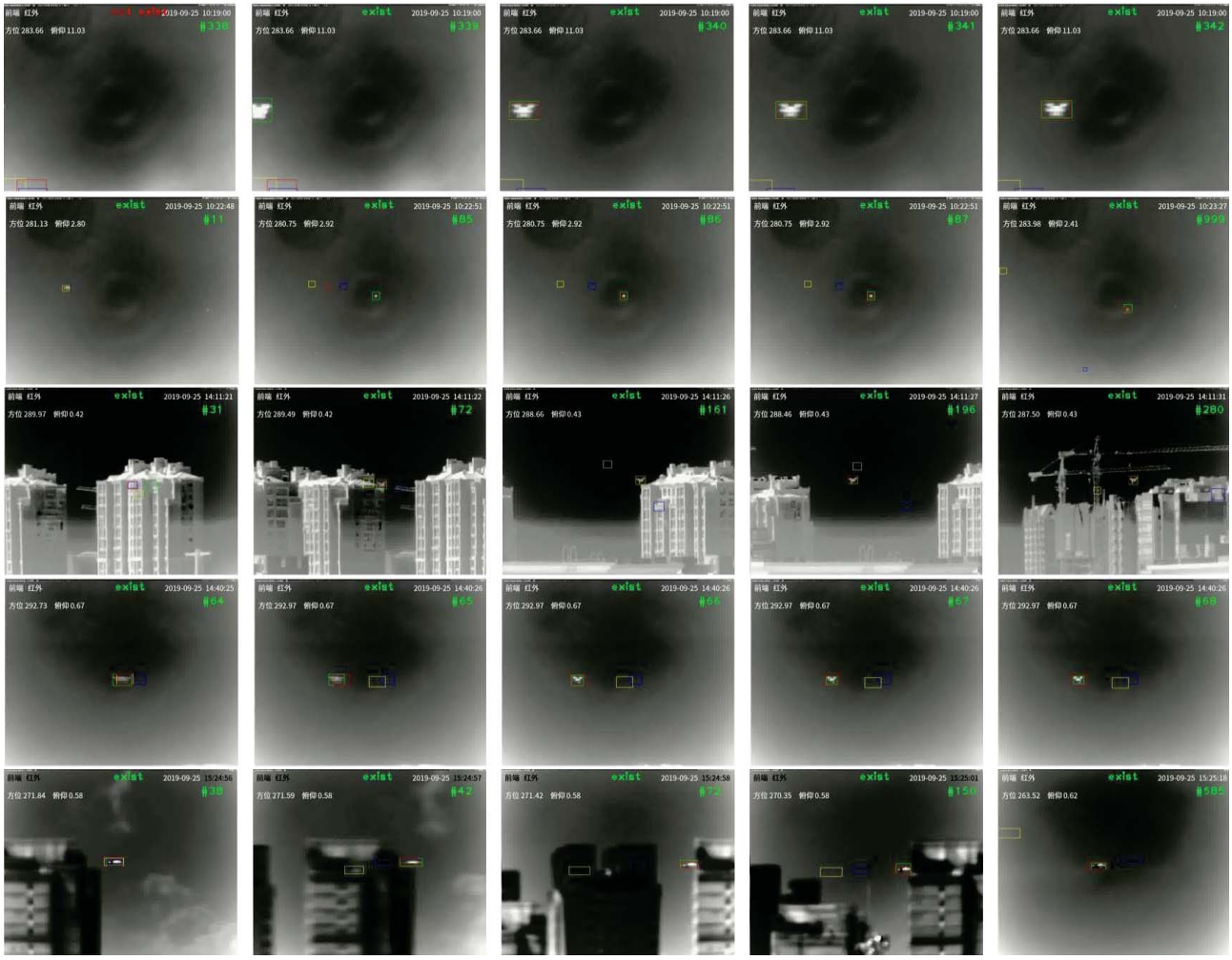

Figure 5. Visual results of our tracker, along with SiamFC [15] and ATOM [13]. The red box denotes ours, the green box denotes ground truth, the yellow box denotes SiamFC and the blue box denotes ATOM. All the data from Anti-UAV infrared dataset. From top to bottom: 20190925_101846_1_1, 20190925_101846_1_7, 20190925_140917_1_4, 20190925_143900_1_3, 20190925_152412_1_2.

The closest algorithm to our tracking method is ATOM. Different from it, we remove the IoU-Net branch, which causing tracking drift once the predicted IoU score is not correct and we add a feature attention mechanism and an expansion search strategy. Our method gets 0.682 acc score which improves 0.11 compared with ATOM.

4.3. Ablation study

We perform an ablation study to demonstrate the impact of each component in the proposed method. We use the same dataset and the evaluation criteria as in Section 4.2. From Table 2. The average tracking speed of our tracking algorithm is 55.6 fps (frames per second). Without feature attention mechanism, the speed is increased to 66.7 fps but the average acc score reduced from 0.682 to 0.669. It means that the backbone network coupled with the attention mechanism can better express the target. Without the expanded search stategy, although the speed is increased to 62.5 fps, the acc score reduced to 0.611, which illustrates the effectiveness of the expansion search strategy to the sudden motion of camera and drone.

4.4. Visual Results

To visualize the performance of our tracker, we provide some representative results of our tracker and the other two baseline methods ATOM [13] and SiamFC [15]. The frames are from the Anti-UAV dataset. As shown in Figure 5, each row represents a video sequence. The red box denotes ours, the green one denotes the ground truth, the yellow box denotes SiamFC and the blue box denotes ATOM. From the first row, at frame 338, the target is out of view. Our algorithm captures the drone at frame 340 as

soon as the drone appears and continues tracking the drone. However, SiamFC and ATOM fail to track once the drone goes out of the view. From the second row, the drone is very small and it appears as a bright blob. At the 85th frame, there is a large-scale displacement on the target, throwing away the tracked boxes of ATOM and SiamFC, but our algorithm can capture the drone at the 86th frame and continue tracking. From the third row, the camera is dragged frequently, and there are buildings in the background, causing interference to the drone. At frame 72, SiamFC and ATOM tracking fail, but our algorithm with feature attention mechanism and extended search mechanism can track targets robustly. From the fourth row, the texture of the drone is not clear. At the 65th frame, the camera suddenly moves, making both SiamFC and ATOM fail in tracking, but our algorithm can track the target well. From the fifth row, all the algorithms can track the drone well until the 38th frame. Then the camera suddenly moves, causing a large-scale displacement of the target position. Only our tracking algorithm can keep tracking the target, and all the other algorithms fail. The same happens also in the 72nd frame, the 150th frame, and the 585th frame, which shows that the strategy of the expansion search proposed is effective.

5. Conclusion

In this paper, a robust and real-time infrared UAVs tracking approach is proposed, which mainly consists in the addition of a feature attention mechanism and an expansion search strategy to a fully convolutional classifier. Experiments on the Anti-UAV dataset show that the proposed infrared tracking algorithm is robust to the challenges in real infrared scenes in real time.

References

[1] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling. Vision meets drones: Past, present and future. arXiv: 2001.06303, 2020.

[2] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. European Conference Computer Vision, 445–461, 2016.

[3] D. P. Young and J. M. Ferryman. PETS Metrics: On-line performance evaluation service. International Conference on Computer Communications and Networks, 317–324, 2005.

[4] M. Felsberg, A. Berg, G. Hager, J. Ahlberg, M. Kristan, A. Leonardis, J. Matas, G. Fernandez, L. Cehovin, and et al. The thermal infrared visual object tracking VOTTIR2015 challenge results. In International Conference Computer Vision workshop proceedings, VOT2015 Workshop, 2015.

[5] M. Felsberg, M. Kristan, J. Matas, A. Leonardis, R.Pflugfelder, G.Hager, A.Berg, A.Eldesokey, J.Ahlberg, L. Cehovin, T. Vojir, A. Lukezic, G. Fernandez, and et al. The thermal infrared visual object tracking VOT-TIR2016 challenge results. In European Conference Computer Vision workshop proceedings, VOT2016 Workshop, 2016.

[6] C. Li, X. Liang, Y. Lu, N. Zhao, and J. Tang. RGB-T object tracking: Benchmark and baseline. Pattern Recognition, 96, 106977, 2019.

[7] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg. Deep motion features for visual tracking. International Conference on Pattern Recognition, 1243-1248, 2016.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. Conference on Computer Vision and Pattern Recognition, 2544-2550, 2010.

[9] 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, 37(3), 583-596, 2015.

[10] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. International Conference on Computer Vision, 4310-4318, 2015.

[11] F. Li, C. Tian, W. Zuo, L. Zhang, and M. H. Yang. Learning spatial-temporal regularized correlation filters for visual tracking. Conference on Computer Vision and Pattern Recognition, 4904-4913, 2018.

[12] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ECO: Efficient convolution operators for tracking. Conference on Computer Vision and Pattern Recognition, 6931-6939, 2017.

[13] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ATOM: Accurate tracking by overlap maximization. Conference on Computer Vision and Pattern Recognition, 2019.

[14] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. European Conference Computer Vision, 2018.

[15] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision, 850-865, 2016.

[16] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. Conference on Computer Vision and Pattern Recognition, 8971-8980, 2018.

[17] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. Conference on Computer Vision and Pattern Recognition, 1328-1338, 2019.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, 91-99, 2015.

[19] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte. Learning discriminative model prediction for tracking. International Conference on Computer Vision, 6182-6191, 2019.

[20] A. Berg. Detection and tracking in thermal infrared imagery. Doctoral dissertation, Linköping University Electronic Press. 2016.

[21] R. Yang, Y. Zhu, X. Wang, C. Li, and J. Tang. Learning target-oriented dual attention for robust RGB-T tracking. International Conference on Image Processing, 3975-3979, 2019.

[22] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan. Synthetic data generation for end-

to-end thermal infrared tracking. IEEE Transactions on Image Processing, 28(4), 1837-1850, 2018.
[23] Y. Gao, C. Li, Y. Zhu, J. Tang, T. He, and F. Wang. Deep adaptive fusion network for high performance RGB-T tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.
[24] L. Zhang, M. Danelljan, A. Gonzalez-Garcia, J. van de Weijer, and F. Shahbaz Khan. Multi-modal fusion for end-to-end RGB-T tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.
[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. Conference on Computer Vision and Pattern Recognition, 770-778, 2016.
[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and A. C. Berg. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252, 2015.
[27] L. Trottier, P. Giguere, and B. Chaib-draa. Parametric exponential linear unit for deep convolutional neural networks. International Conference on Machine Learning and Applications, 207-214, 2017.
[28] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994.
[29] H. Nam, and B. Han. Learning multi-domain convolutional neural networks for visual tracking. Conference on Computer Vision and Pattern Recognition, 4293-4302, 2016.
[30] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. European Conference on Computer Vision, 103-119, 2018.
[31] S. Ioffe, and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, 448-456, 2015.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097-1105, 2012.
[33] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. Conference on Computer Vision and Pattern Recognition, 7132-7141, 2018.
[34] S. Woo, J. Park, J. Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. European Conference on Computer Vision, 3-19, 2018.
[35] P. Chen. GridMask: Data augmentation. arXiv preprint arXiv:2001.04086, 2020.
[36] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg. Unveiling the power of deep tracking. European Conference on Computer Vision, 483-493, 2018.
[37] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. Conference on Computer Vision and Pattern Recognition, 4282-4291, 2019.