A Unified Approach for Tracking UAVs in Infrared.

Jinjian Zhao *, Xiaohan Zhang *, and Pengyu Zhang †

School of Information and Communication Engineering, Dalian University of Technology, China
{ 732514581, zxiaohan, pyzhang}@mail.dlut.edu.cn

Abstract

With complex camera and object movement, the tracked object often suffers camera motion, out of view, dramatic scale variation, etc., which severely influence tracking performance. Due to the fast speed and tiny size of unmanned aerial vehicles(UAV), it is crucial to design a robust framework for tracking UAVs. This paper carefully designs a unified framework, including a local tracker, camera motion estimation module, bounding box refinement module, re-detection module and model updater. The camera motion estimation module achieves motion compensation for the local tracker. Then, the bounding box refinement module aims to measure an accurate bounding box. If the target is missing, we switch to the re-detection module to re-localize the target when it reappears. We also adopt a model updater to control the updating process and filter out unreliable samples. Numerous experimental results on 9 visual/thermal datasets show the effectiveness and generalization of our framework.

1. Introduction

Object tracking is an important task in computer vision, which has drawn the great attention of many researchers in the past few decades. Given one video sequence and the initial state of an arbitrary target, object tracking aims to predict the position and scale of the target in each frame. Tracking has made great progress recently, where many outstanding trackers are proposed. SuperDiMP combines the bounding-box regressor of PrDiMP [10] with the standard classifier of DiMP [3] equipping tracker with better robustness and scale estimation. LTMU [7] proposed a meta updater to control the update of online-updated short-term trackers. It avoids polluting useful appearance information in long-term videos with frequent target disappearance. JMAC [46] jointly models motion and appearance modules to overcome the sudden shaking in videos. All the above methods try to overcome the challenges in video sequences from different aspects.

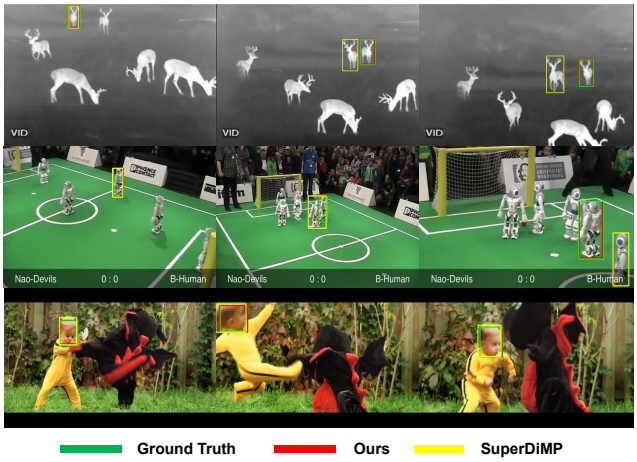

Figure 1. Examples of our framework on three different tracking benchmarks. 'Ours' means our framework, 'SuperDiMP' means competing method. The frames of three rows are from LSOTB-TIR [29], LaSOT [12] and OTB [41] benchmarks.

Motivated by the above analysis, we find that the tracked objects often suffer various challenges such as camera motion, out of view, dramatic scale variation, especially for tracking UAVs. In this work, we design a unified framework shown in Figure2 to handle the above problems. To achieve this goal, we firstly design a camera motion estimation module, which reset the search region of the local tracker to solve the sudden camera motion. Secondly, we adopt a bounding box refinement module, which refines the local tracker's outputs, to estimate the target scale efficiently. Then, we introduce a re-detection module into the unified object tracking framework to settle the problem of the target disappearance. Finally, we adopt a model updater to control the tracker's online update. Numerous experi-

*Equal contribution. Author ordering determined by names.
†Corresponding author.

ments on nine different benchmarks illustrate the effectiveness and generalization of our framework.

The main contributions of this work can be summarized as follows:

- We ensemble the local tracker, the camera motion estimation module, the bounding box refinement module, the re-detection module and the model updater into a unified framework to solve camera motion, out-of-view and dramatic scale variation in complicated scenarios.
- Numerous experiments on ICCV2021 Anti-UAV Challenge test-dev subset, LSOTB-TIR, OTB2015, GOT-10k, TrackingNet, NFS, UAV123, LaSOT, VOT2019LT benchmarks show the effectiveness of the proposed framework.

2. Related Work

2.1. Single Object Tracking

Although the traditional correlation filter object tracking methods [4, 16, 17, 11] were widely used in previous tracking frameworks, most of them were not very accurate comparatively due to the lack of robust feature representation. Recent methods based on Siamese network had drawn great attention in the domain of single object tracking. The pioneering work, namely SiamFC [2], exposed the advantage of the Siamese network by introducing similarity learning into tracking. The Siamese network trained a network with pairs comprising exemplar images and search images in the offline phase. During tracking, the features extracted from the template image and the search image were correlated to compute the similarity score map. Many improvements based on Siamese network had been made. SiamRPN [24] introduced the Region Proposal Network (RPN) [36] into Siamese network to obtain accurate bounding boxes. DaSiamRPN [48] introduced a training sampling strategy and designed a distractor-aware module to improve the discrimination power of SiamRPN. SiamRPN++ [23] proposed a deeper Siamese tracker by applying an effective spatial-aware sampling strategy. SiamFC++ [42] proposed an anchor-free tracker with significant performance gain since the traditional tracking method based on pre-defined anchors seriously suffered from the hyperparameters. A key limitation of Siamese network was ignoring the background appearance information during inference. DiMP [3] exploited the appearance information of both target and background by introducing a discriminative tracking architecture. PrDiMP [10] modeled the label noises and introduced a probabilistic representation into DiMP. SuperDiMP introduced the PrDiMP bounding-box regressor into DiMP to obtain a more reliable classification and regression output.

Unlike general tracking, various special challenges (e.g., severe occlusion, fast motion) are involved in tracking UAVs [5]. The study of tracking UAVs advanced rapidly in recent years. Recent works [39, 40, 22, 5] attempted to improve the tracking framework according to the characteristics of UAVs. S-Siam [39] aimed to deal with the problem of small targets and fast motion by adjusting the camera adaptively. A framework [40] proposed a feature attention module and a target searching strategy for tracking drones. ADTrack [22] proposed an anti-dark UAV tracker by integrating a low-light image enhancer into a CF-based tracker. [5] proposed a real-time attentional Siamese tracker for tracking UAVs. Furthermore, various benchmarks (e.g., AntiUAV [1], UAVDark [22]) for tracking UAVs have been published.

2.2. Thermal infrared object tracking

Comparing with cameras for the visual spectrum, cameras for thermal are capable of operating complex situations including darkness, shadow and illumination changes [13]. Thus, tracking based on thermal infrared (TIR) is also important when the visual spectrum information is missing. Different from ray-scale visual imagery tracking, the blooming and lower resolution problems are much more frequent in thermal infrared tracking [1]. HSSNet [26] proposes a hierarchical spatial-aware siamese network for TIR tracking. MLSSNet [28] develops a multi-level similarity model for handling distractors in TIR tracking. MMNet [27] proposes a multi-task matching framework that integrates the TIR-specific features and the fine-grained correlation features. ABCD [1] proposed an adaptive object region and background weighted scaled channel coded distribution field method for short-term single-object thermal infrared tracking. Recently, a robust and real-time tracking algorithm for infrared drones is proposed by [40]. Furthermore, many large-scale benchmarks (e.g., VOT-TIR2015 [13], LSOTB-TIR [29]) for thermal infrared tracking have been published.

2.3. Re-Detection in Visual Tracking

Due to the common problem of target frequent disappearances and reappearances, the re-detection module is widely used in long-term tracking frameworks. The pioneering work Tracking-Learning-Detection (TLD) [20] was designed as two parts: a local tracker and a re-detection module. By exploiting the core idea of re-detection, some further improvements [32, 47, 44, 18, 7] were made. MBMD [47] proposed a SiamPRN-based long-term framework, which re-detect the target with a sliding window when the verification network distrusted the candidates generated by the local tracker. SPLT [44] used a skimming module to choose possible regions among many sliding windows so that the re-detection module could achieve a good speed-accuracy trade-off. Global Track [18], which

[1] Provided by https://anti-uav.github.io/ .

was designed based on the two-stage detector, was developed to perform pure global instance searching without any trajectory refinement. LTMU [7] proposed a meta-updater module and designed a long-term framework that applied the Global Track as the re-detection module.

2.4. Camera Motion Model

Since the development of the neural network, the field of object tracking has grown rapidly. However, camera motion estimation has not been paid enough attention in recent researches. Kernel-based object tracking [6] proposed a new approach for target representation and localization. It successfully coped with camera motion situations by integrating with motion filters and data association techniques. A camera motion estimation method [25] is proposed for drone tracking by applying geometric transformation based on background SURF feature points. The camera motion estimation is used to guide the search area for tracking roughly, and it is very effective for handling fast camera motion in drone tracking. JMMAC [46] proposed an RGB-T tracking framework based on ECO [8]. It comprehensively considered the result from ECO tracker and the result predicted by target and camera motion cues to give the final result. TrackletNet Tracker (TNT) [37] compensated camera movement between adjacent frames by proposing the EG-IOU module based on Epipolar Geometry.

3. Method

In this section, we describe the proposed framework for tracking. Our framework consists of five modules: Camera Motion Estimation (CME) module, Local Tracker (LT), Bounding Box Refinement (BBR) module, Re-Detection (RD), Model Updater (MU). The SuperDiMP [2] method is adopted as the local tracker due to its effectiveness. The overall framework is presented in Figure 2.

The CME models camera movement in each frame and gives a reliable search region by comparing the current frame with the reference frame selected from the past. The local tracker exports the target bounding box from the search region. The BBR takes the result of the local tracker as input and outputs a more accurate bounding box. The RD module detects the target when the local tracker misses the tracked object. Additionally, we use the MU module to control the update of trackers. Our framework can improve the robustness of the base tracker under complicated conditions in various modalities and scenes by such careful design.

3.1. Camera Motion Estimation Module

As for sudden camera motion, it is difficult to predict the location of the target. Therefore, we propose the CME to reset the search region of the local tracker based on image registration, where the search region of the reference frame is mapped into the current frame. Since the modeling of 3D solid is difficult, we assume the depth difference can be ignored and all objects are in a 2D plane. In CME, we firstly extract the scale-invariant feature transform (SIFT) [31] key points of the reference and current frames. In our experiment, we select the latest reliable image from the latest 10 frames, which is measured by a verifier motioned later as the reference frame. Note that, since the re-detection module outputs a discontinuous target trajectory, we remove the previous frames which utilize the global tracker's results as final results. Then, we match their key points followed by an outlier removal method (RANSAC) and obtain a transformation matrix $\boldsymbol{O}$ to model camera movement. Finally, the search region of the reference frame R_{t_r} is mapped into the current frame by the obtained transformation matrix $\boldsymbol{O}$ as shown in Figure 3, which provides a sketch map of search region mapping based on CME. The search region of the current frame R_t can be obtained by

$$R_t = \mathcal{T}(R_{t_r}; \boldsymbol{O}) \tag{1}$$

where $\mathcal{T}(.; \boldsymbol{O})$ denotes the transformation function with the parameter transformation matrix $\boldsymbol{O}$. In the experiment. We adopt affine transformation as $\mathcal{T}$.

In this way, the proposed CME introduces the stable search region into the local tracker and leads to a robust object tracking.

3.2. Bounding Box Refinement Module

In our framework, We adopt SuperDiMP as the local tracker, which combines the classifier of DiMP [3] and the bounding box regressor of PrDiMP [10]. However, the bounding box regressor cannot give accurate bounding boxes with low resolution and low contrast frames. Inspired by the multiple-stage tracking strategy, we attempt to address this dilemma in two steps: rough positioning by the local tracker and location refinement by the bounding box refinement module. AlphaRefine [43] is a plug-and-play module with a strong regression capability to refine the local tracker's output efficiently. We adopt Alpha-Refine Module as BBR in our framework due to its flexibility and effectiveness.

As shown in Figure 4, the BBR can be divided into four steps: (i) Extend the coarse result of the local tracker into a concentric search region. (ii) Extract the features from the obtained search region of the search frame and the template region of the first frame with a parameter-shared backbone. (iii) Fuse the obtained features with the feature fusion layer. (iv) Regress the box coordinate with the fused feature map by the bounding box regressor. Specifically, the pixel-wise correlation is adopted as the feature fusion layer, and the

[2]The pre-trained model of SuperDiMP is provided by https://github.com/visionml/pytracking.

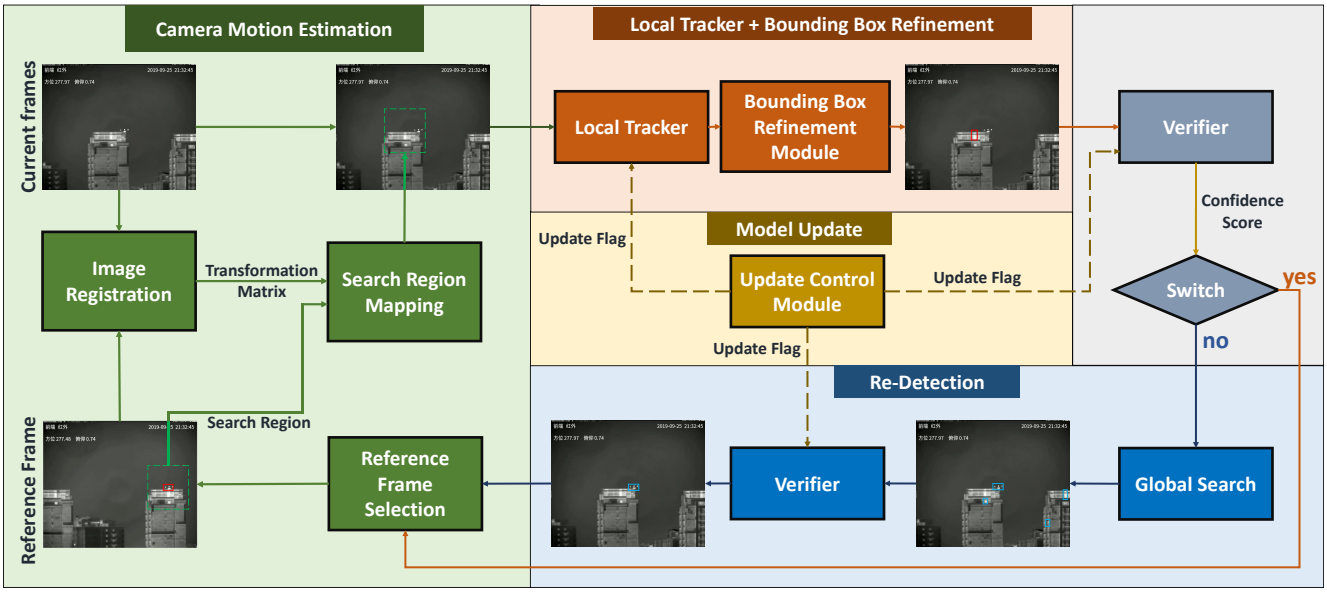

Figure 2. Overall framework of our method. Better viewed in color with zoom-in.

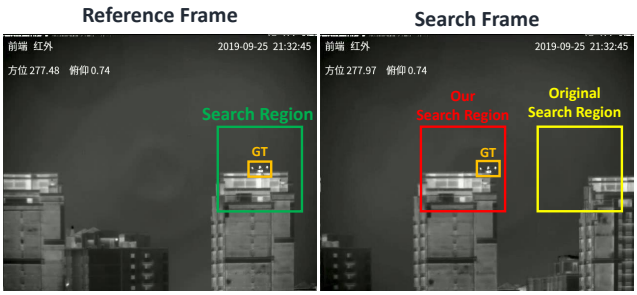

Figure 3. Search region mapping based on CME.

corner head is adopted as the bounding box regressor to predict the top-left corner and the bottom-right corner directly.

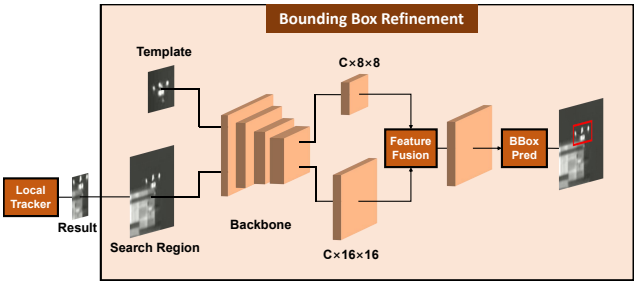

Figure 4. The architecture of bounding box refinement module (AlphaRefine [43]). Better viewed in color with zoom-in.

In this way, the proposed BBR introduces the strong regression capability into the framework and leads to accurate object tracking.

3.3. Re-Detection module

When the target is out of view or suffers background cluttering, the local tracker is fragile to detect the target. To solve this issue, we adopt the RD to locate the target when the target reappears.

However, the inappropriate re-detection always leads to the distractor problem. It is essential to re-detect the target only if the local tracker losses it. We borrow the re-detection scheme from long-term tracking. We adopt MDNet [35] as the verifier in our work. The verifier evaluates the correctness of the local tracker's result in each frame and gives a confidence score. A switcher is applied to monitor the obtained confidence score: if the confidence score is lower than the threshold for continuous five frames, the RD will be activated.

As the RD is activated, we adopt the Global Track [18] method to give the possible candidates. The Global Track method is a global instance searching method without any locality assumption or temporal consistency assumption. Specifically, the features are extracted from the query frame (the first frame) and the search frame (the whole image of the current frame) by the backbone, and the convolution operator is applied to generate the query-specific object candidates. Then the Query-Guided RCNN network classifies and refines the obtained candidates. The top-K candidates, which are sorted by the classification scores, are retained. K is set to 5 in our experiment.

We weed out the inappropriate candidates by imposing prior information(e.g., area, aspect ratio). Then, every candidate is given a confidence score by the verifier. The candidate with the highest confidence score is regarded as the

final output. Incidentally, once the RD gives a result, we will reset the search region of the local tracker. The proposed RD is listed in Algorithm 1.

Algorithm 1: Re-Detection module

Input: The ground truth of query frame $\boldsymbol{g}_q$, the image of query frame $\boldsymbol{I}_q$, the image of search frame $\boldsymbol{I}_s$, the threshold $t_u^A, t_l^A, t_u^R, t_l^R, t^S$

Output: Bounding Box $\boldsymbol{b}$

Calculate the area $a_q \in \mathbb{R}^1$ of $\boldsymbol{g}_q$;
Calculate the aspect ratio $r_q \in \mathbb{R}^1$ of $\boldsymbol{g}_q$;
Generate K candidates $\{C_i\}_{i=1}^K$ via $\boldsymbol{I}_s$, $\boldsymbol{I}_q$, $\boldsymbol{g}_q$ using Global Track [18];
for $i = 1, 2, 3, ..., K$ **do**
  Calculate the area a_c of C_i;
  if $a_c/a_q > t_u^A$ *or* $a_c/a_q < t_l^A$ **then**
    $C \leftarrow C \backslash C_i$;
  end
  Calculate the aspect ratio r_c of C_i;
  if $r_c/r_q > t_u^R$ *or* $r_c/r_q < t_l^R$ **then**
    $C \leftarrow C \backslash C_i$;
  end
  Evaluate the confidence score s_i of C_i by verifier;
  if $s_i < t^S$ **then**
    $C \leftarrow C \backslash C_i$;
  end
end
if $C \neq \varnothing$ **then**
  $idx \leftarrow \arg\min_i \ s_i$;
  $\boldsymbol{b} \leftarrow C_{idx}$;
else
  $\boldsymbol{b} \leftarrow \varnothing$;
end
return $\boldsymbol{b}$

3.4. Model Updater

In our framework, the local tracker and the verifier need to be updated throughout the tracking process. However, an inappropriate update may lead to unstable tracking. We adopt MU (borrowed from Meta-Updater [7]) to tell whether the trackers need to be updated in each frame. It takes into account discriminative, geometric and appearance cues and gives an effective marker for the determination of update with an offline trained cascaded LSTM module. This section introduces the update details in three parts: Meta-Updater, update details for local tracker and update details for verifier.

Meta-Updater. It is essential to encode the important cues into vectors for the cascaded LSTM module. For geometric cues, MU utilizes the temporal variation of the bounding box to represent the motion information regarding the target. The bounding box at t-th frame is denoted as $\boldsymbol{b}_t$. For discriminative cues, MU use the response map $\boldsymbol{M}_t$ of the local tracker (at t-th frame) to represent the discriminative information. A confidence score and a response vector is defined to represent the $\boldsymbol{M}_t$. To be specific, the confidence score s_t^C can be obtained by

$$s_t^C = max(\boldsymbol{M}_t) \tag{2}$$

and the response vector $\boldsymbol{v}_t^R$ can be obtained by

$$\boldsymbol{v}_t^R = f^R(\boldsymbol{M}_t; \boldsymbol{W}^R) \tag{3}$$

where $f^R(.; \boldsymbol{W}^R)$ denotes the CNN model with the parameter $\boldsymbol{W}^R$. For appearance cues, the appearance score is applied in MU which measures the difference between the template region at the first frame $\boldsymbol{I}_0$ and tracked result at the t-th frame $\boldsymbol{I}_t$. The appearance score s_t^A can be obtained by

$$s_t^A = ||f^A(\boldsymbol{I}_t, \boldsymbol{W}_A) - f^A(\boldsymbol{I}_0, \boldsymbol{W}_A)||_F \tag{4}$$

where $f^R(.; W^R)$ denotes the CNN model based on ResNet-50 [15] with the parameter $\boldsymbol{W}^R$.

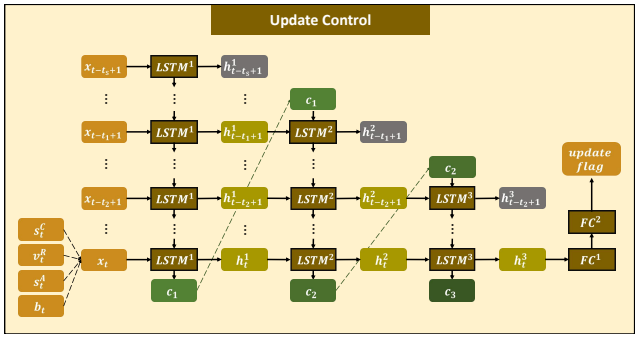

Figure 5. The architecture of model update (Meta-updater [7]). Better viewed in color with zoom-in.

As shown in Figure 5, in each frame, the obtained $\boldsymbol{b}_t$, s_t^C, $\boldsymbol{v}_t^R$, s_t^A are contacted into a vector, denotes as $\boldsymbol{x}_t$. The obtained sequential vectors $\boldsymbol{x}_{t-t_s+1}, ..., \boldsymbol{x}_{t-1}, \boldsymbol{x}_t$ are sent into the three-stage cascade LSTM network. The output of the cascade LSTM $\boldsymbol{h}_t^3$ is further processed by two fully connected layers to obtain the binary update flag.

Update details for local tracker. According to the original update strategy of DiMP[3], the local tracker generates a set of samples by data augmentation in the first frame to initialize the classifier. During tracking, the training samples are collected when the local tracker gives a high confidence score, and the classifier is updated every 20 frames and every distractor. However, the MU is applied to decide if we should update the classifier. During tracking, whenever MU gives an update score higher than the threshold, the training

samples are collected and the classifier is updated. Incidentally, the limit of the size of training samples set is set to 50.

Update details for the verifier. MDNet [35] is adopted as the verifier. The original update strategy of MDNet is divided into two parts: short-term update whenever the confidence score is less than the threshold and long-term update at every fixed interval. By applying MU , the positive features and the negative features of a frame will be collected when a high update score is given so that the verifier can be updated at every fixed interval with the collected features.

4. Experiments

4.1. Evaluation on thermal infrared benchmarks.

We evaluate the tracker on ICCV2021 Anti-UAV Challenge test-dev, LSOTB-TIR [29] and other 7 long/short term datasets, which show the effectiveness and generalization of our framework in different scenarios.

ICCV2021 Anti-UAV Challenge test-dev subset. The ICCV2021 Anti-UAV Challenge dataset, containing test-dev subset and test-challenge subset, covers multiple occurrences of multi-scale UAVs. The test-dev subset contains 140 thermal infrared video sequences, which involve fast motion, target disappearance, and many other challenging scenarios. We follow its metrics to calculate the tracking accuracy of our tracker and some other influential algorithms, which include LTMU [7], SiamRPN++ [23] and SuperDiMP. Table 1 shows the performance of these methods.

Table 1. Evaluation of our tracker and other algorithms on the ICCV2021 Anti-UAV Challeng test-dev subset.

Tracker	Ours	LTMU	SuperDiMP	SiamRPN++
Acc	0.670	0.605	0.559	0.403

Results on LSOTB-TIR Benchmark. The LSOTB-TIR dataset [29] contains 1400 sequences (1280 for training and 120 for evaluation) with 47 object classes and more than 600K frames. It is a recently more diverse thermal infrared tracking dataset. We follow the one-pass evaluation rules and use success and precision plots to evaluate our tracker on the evaluation dataset. The evaluation dataset includes 12 challenges such as fast motion, scale change, out of view, deformation, etc., figure 6 shows the success and precision plots of our tracker and some competitive RGB or thermal infrared algorithms [11, 26, 30, 28, 38, 8, 45, 7] in the LSOTB-TIR toolkit. In Figure 6, our tracker achieves the best result which illustrates the effectiveness of our method.

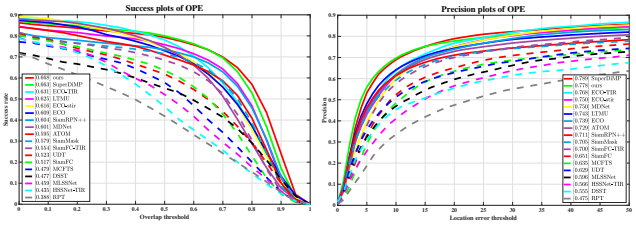

Figure 6. The success and precision plots of different trackers on LSOTB-TIR evaluation dataset. Better viewed in color with zoom-in.

4.2. Evaluation on short-term benchmarks.

In this section, we also evaluate our tracker in visual tracking dataset to show its generalization. We compare our method with other outstanding RGB trackers in 5 popular datasets, including NFS, OTB2015, UAV123, GOT-10k, TrackingNet.

Results on NFS Benchmark. Figure 7 reports representative tracking results on the NFS benchmark. The NFS dataset [14] contains 100 sequences with an average length of 479 frames. We follow the one-pass evaluation rules and use success and precision scores to evaluate our tracker on the 30fps version. Our method achieves a success score of 66.1%, which is comparable to recent state-of-the-art trackers.

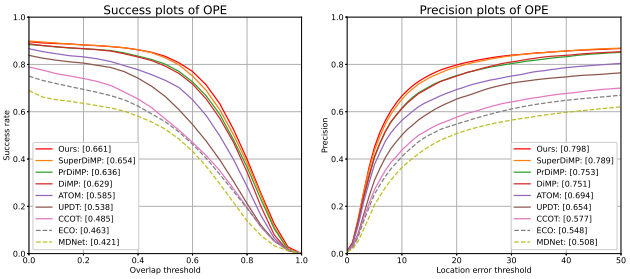

Figure 7. Success and precision plots of different methods on the NFS dataset. Better viewed in color with zoom-in.

Results on OTB2015 Benchmark. We compare our tracker with other state-of-the-art algorithms on the OTB2015 dataset [41], which contains 100 videos with an average length of 590 frames, and calculate the success and precision scores over varying overlap thresholds. Table 2 reports the performance of different methods, it shows that our framework achieves comparable results with other state-of-the-art short-term trackers.

Results on UAV123 Benchmark. The UAV123 dataset [33] contains 123 sequences with an average length of 915 frames. We follow the one-pass evaluation rules and use success and precision plots to evaluate our tracker with SuperDiMP, PrDiMP [10], DiMP [3], ATOM [9] and etc., our

Table 2. Comparison of our tracker with other algorithms on OTB2015. The best three methods are shown in red, green and blue, respectively. The table is arranged from top to bottom according to Success score.

Method	Success	Precision
Ours	0.709	0.913
SuperDiMP	0.701	0.910
SiamRPN++	0.696	0.915
ECO	0.691	0.910
DiMP-50	0.687	0.899
MDNet	0.678	0.909
Ocean-online	0.684	0.920
ATOM	0.671	0.882
DaSiamRPN	0.658	0.878

method achieves the best success score of 69.7% among the above outstanding trackers, details shown in Figure 8.

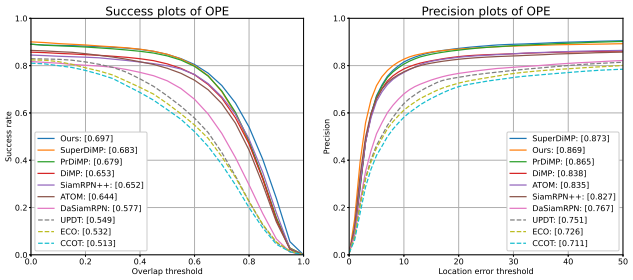

Figure 8. Success and precision plots of different methods on the UAV123 dataset. Better viewed in color with zoom-in.

Results on GOT-10k Benchmark. As a large-scale tracking dataset, the GOT-10k [19] dataset contains 10k videos for training and 180 videos for testing. There is no class intersection between them. We follow the protocol of GOT-10k and submit the tracking results to the official evaluation server. Then, we compare our tracker with some outstanding trackers. Table 3 shows that our method reaches an AO of 69.2% surpassing SuperDiMP by 1.9%.

Results on TrackingNet Benchmark. The test set of the TrackingNet dataset [34] contains 511 challenging sequences with an average length of 442 frames. we submit our tracking results to the official evaluation server and report the tracker's Success, Normalized Precision(P_{Norm}) and Precision scores with other competitive methods, details shown in Figure 4.

4.3. Evaluation on long-term benchmarks.

We also evaluate the proposed method on long-term tracking benchmarks, including LaSOT [12] and VOT2019LT [21] datasets.

Results on LaSOT Benchmark. The LaSOT dataset is one

Table 3. Comparison of our tracker with some outstanding algorithms on Got-10k benchmark. The best three methods are shown in red, green and blue, respectively. The table is arranged from top to bottom according to AO.

Tracker	AO	$SR_{0.5}$	$SR_{0.75}$
Ours	0.692	0.797	0.601
SuperDiMP	0.673	0.787	0.592
PrDiMP	0.634	0.738	0.543
Ocean	0.611	0.721	0.473
DiMP-50	0.611	0.717	0.492
siamFC++	0.595	0.695	0.479
ATOM	0.556	0.634	0.402
siamRPN++	0.517	0.616	0.325
SiamFC	0.348	0.353	0.098
ECO	0.316	0.309	0.111
MDNet	0.299	0.303	0.099

Table 4. Comparison of our tracker with some competitive methods on the TrackingNet dataset. The best three methods are shown in red, green and blue, respectively. The table is arranged from top to bottom according to Success score.

Tracker	Success	P_{Norm}	Precision
Ours	79.3	84.4	76.2
SuperDiMP	77.9	83.4	73.2
PrDiMP	75.8	81.6	70.4
MAML	75.7	82.2	72.5
SiamFC++	75.4	80.0	70.5
KYS	74.0	80.0	68.8
DiMP	74.0	80.1	68.7
SiamRPN++	73.3	80.0	69.4
D3S	72.8	76.8	66.4
ATOM	70.3	77.1	64.8
MDNet	60.6	70.5	56.5
SiamFC	57.1	66.3	53.3
ECO	55.4	61.8	49.2

of the most recent large-scale dataset with high-quality annotations. It consists of 1400 challenging sequences which include 1120 training sequences and 280 testing sequences. The average length of every sequence is about 2500 frames. We follow the one-pass evaluation rules and use success and precision scores to compare our tracker with other outstanding algorithms. Our tracker achieves a success score of 0.659 surpassing SuperDiMP by 2.1%, Figure 9 shows the performance of some representative algorithms.

Attribute-based evaluation on LaSOT Benchmark. Figure 10 reports the attribute-based evaluation of some competitive trackers which contain SuperDiMP, Ocean, ATOM, DiMP, SiamRPN++, RTMDNet, and SiamFC. It shows that in out-of-view, fast motion, scale variation, camera motion,

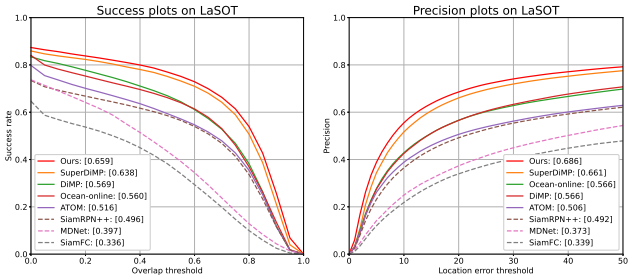

Figure 9. Success and precision plots of different methods on the LaSOT dataset. Better viewed in color with zoom-in.

and partial occlusion situations, our tracker performs better than the base tracker SuperDiMP.

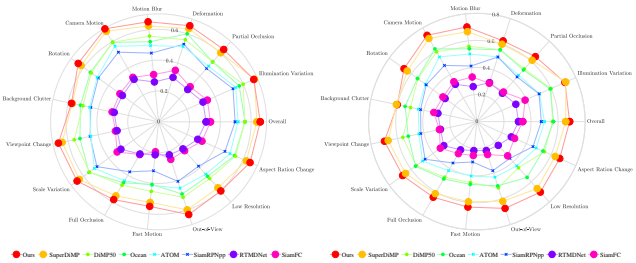

Figure 10. Success and Precision scores of different attributes on the LaSOT test dataset. Better viewed in color with zoom-in.

Results on VOT2019LT benchmark. As a long-term dataset, the VOT2019LT dataset contains 50 sequences with an average length of 4305 frames. There are more challenges since some uncommon objects and more out of view conditions in it. We follow the evaluation protocol and report the Precision, Recall, and F1 scores of some competitive algorithms. The performance of trackers is shown in Table 5.

4.4. Ablation Study

In this section, we conduct ablation analysis of our framework on the ICCV2021 Anti-UAV Challenge test-dev dataset.

Effectiveness of different parts of our framework. We evaluate Camera Motion Estimation Module (CME), Bounding box Refinement Module (BBR), Re-Detection Module (RD) and Model Updater (MU), each module of our framework acts an important role. The results are shown in Table 6, showing the contribution of different modules to our framework.

Impact of different numbers of global detection boxes. As an important hyperparameter, different numbers of global detection boxes may affect the overall performance. We investigate the appropriate number of it. Table 7 shows that the effect of different sets is small. Therefore, in this work, we set the number of proposals to be 5.

Table 5. Comparison of our tracker with competitive algorithms on VOT2019LT benchmark. The best three methods are shown in red, green and blue, respectively. The table is arranged from top to bottom according to F1 score.

Tracker	F1	Precision	Recall
Ours	0.708	0.726	0.691
LTMU	0.697	0.721	0.674
CLGS	0.674	0.739	0.619
SiamDW_LT	0.665	0.697	0.636
SuperDiMP	0.663	0.671	0.656
mbdet	0.567	0.609	0.530
SiamRPNsLT	0.556	0.749	0.443
Siamfcos-LT	0.520	0.493	0.549
CooSiam	0.508	0.482	0.537
ASINT	0.505	0.517	0.494
FuCoLoT	0.411	0.507	0.346

Table 6. The contribution of every module in our framework.

Tracker	+MU	+RD	+BBR	+CME	Acc (%)
SuperDiMP					55.9
	✓				57.0 (+1.1)
	✓	✓			59.3 (+2.3)
	✓	✓	✓		64.6 (+5.3)
	✓	✓	✓	✓	67.0 (+2.4)

Table 7. The impact of different numbers of global detection boxes.

Num. Boxes	5	10	20	30	40
Acc	0.670	0.662	0.662	0.663	0.662

5. Conclusion

To solve the challenging issues in anti-UAV tracking, including camera motion, out of view and scale variation, we propose a unified tracking framework for visual, thermal infrared, and long-term tracking. First, we apply motion compensation caused by camera movement via the camera motion estimation module. Then, we utilize the re-detection mechanism to detect and handle the case, when the target moves out of view. Finally, we adopt an accurate box regression module to obtain a precise scale estimation. Numerous experiments on Anti-UAV challenge, LSOTB-TIR, OTB2015, GOT10k, NFS, UAV123, TrackingNet, LaSOT and VOT2019LT shows strong potential of our framework in handling visual/thermal, short-term and long-term tracking scenes, which can be widely used in real world applications.

References

[1] Amanda Berg. *Detection and Tracking in Thermal Infrared Imagery*. PhD thesis, Linköping University, 2016. 2

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016. 2

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6181–6190, 2019. 1, 2, 3, 5, 6

[4] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *ECCV*, pages 483–498, 2018. 2

[5] Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. SiamAPN++: Siamese attentional aggregation network for real-time uav tracking. *CoRR*, abs/2106.08816, 2021. 2

[6] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *TPAMI*, 25(5):564–577, 2003. 3

[7] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, pages 6297–6306, 2020. 1, 2, 3, 5, 6

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017. 3, 6

[9] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *CVPR*, pages 4655–4664, 2019. 6

[10] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, pages 7181–7190, 2020. 1, 2, 3, 6

[11] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, pages 1–11, 2014. 2, 6

[12] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. LaSOT: A high-quality large-scale single object tracking benchmark. In *CVPR*, pages 5369–5378, 2019. 1, 7

[13] Michael Felsberg, Amanda Berg, Gustav Hager, Jorgen Ahlberg, Matej Kristan, Jiri Matas, Ales Leonardis, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, et al. The thermal infrared visual object tracking VOT-TIR2015 challenge results. In *ICCVW*, pages 76–88, 2015. 2

[14] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, pages 1134–1143, 2017. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[16] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715, 2012. 2

[17] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2014. 2

[18] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Globaltrack: A simple and strong baseline for long-term tracking. In *AAAI*, pages 11037–11044, 2020. 2, 4, 5

[19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 43(5):1562–1577, 2021. 7

[20] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2011. 2

[21] Matej Kristan, Amanda Berg, Linyu Zheng, and et al. The seventh visual object tracking VOT2019 challenge results. In *ICCVW*, pages 2206–2241, 2019. 7

[22] Bowen Li, Changhong Fu, Fangqiang Ding, Junjie Ye, and Fuling Lin. ADTrack: Target-aware dual filter learning for real-time anti-dark UAV tracking. *CoRR*, abs/2106.02495, 2021. 2

[23] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4277–4286, 2019. 2, 6

[24] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 2

[25] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, pages 4140–4146, 2017. 3

[26] Xin Li, Qiao Liu, Nana Fan, Zhenyu He, and Hongzhi Wang. Hierarchical spatial-aware siamese network for thermal infrared object tracking. *KBS*, 166:71–81, 2019. 2, 6

[27] Qiao Liu, Xin Li, Zhenyu He, Nana Fan, Di Yuan, Wei Liu, and YongSheng Liang. Multi-task driven feature models for thermal infrared tracking. In *AAAI*, pages 11604–11611, 2020. 2

[28] Qiao Liu, Xin Li, Zhenyu He, Nana Fan, Di Yuan, and Hongpeng Wang. Learning deep multi-level similarity for thermal infrared object tracking. *TMM*, 23:2114–2126, 2021. 2, 6

[29] Qiao Liu, Xin Li, Zhenyu He, Chenlong Li, Jun Li, ZiKun Zhou, Di Yuan, Jing Li, Kai Yang, Nana Fan, and Feng Zheng. LSOTB-TIR: A large-scale high-diversity thermal infrared object tracking benchmark. In *ACMMM*, pages 3847–3856, 2020. 1, 2, 6

[30] Qiao Liu, Di Yuan, and Zhenyu He. Thermal infrared object tracking via siamese convolutional neural networks. In *SPAC*, pages 1–6, 2017. 6

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3

[32] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *ICCV*, pages 5388–5396, 2015. 2

[33] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *ECCV*, pages 445–461, 2016. 6

[34] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale

dataset and benchmark for object tracking in the wild. In *ECCV*, pages 310–327, 2018. 7

[35] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016. 4, 6

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2016. 2

[37] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *ACMMM*, pages 482–490, 2019. 3

[38] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, pages 1308–1317, 2019. 6

[39] Zixuan Wang, Zhicheng Zhao, and Fei Su. Real-time tracking with stabilized frame. In *CVPRW*, pages 1028–1029, 2020. 2

[40] Han Wu, Weiqiang Li, Wanqi Li, and Guizhong Liu. A real-time robust approach for tracking UAVs in infrared videos. In *CVPRW*, pages 1032–1033, 2020. 2

[41] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 1, 6

[42] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, pages 12549–12556, 2020. 2

[43] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-Refine: Boosting tracking performance by precise bounding box estimation. In *CVPR*, pages 5289–5298, 2021. 3, 4

[44] Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'skimming-perusal' tracking: A framework for real-time and robust long-term tracking. In *ICCV*, pages 2385–2393, 2019. 2

[45] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Synthetic data generation for end-to-end thermal infrared tracking. *TIP*, 28(4):1837–1850, 2019. 6

[46] Pengyu Zhang, Jie Zhao, Chunjuan Bo, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Jointly modeling motion and appearance cues for robust RGB-T tracking. *TIP*, 30:3335–3347, 2021. 1, 3

[47] Yunhua Zhang, Dong Wang, Lijun Wang, Jinqing Qi, and Huchuan Lu. Learning regression and verification networks for long-term visual tracking. In *arXiv preprint arXiv:1809.04320*, pages 1–8, 2018. 2

[48] Zheng Zhu, Qiang Wang, Li Bo, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 103–119, 2018. 2