SiamCDA

Zhang, Tianlu; Liu, Xueru; Zhang, Qiang; Han, Jungong

Published in:
IEEE Transactions on Circuits and Systems for Video Technology

DOI:
10.1109/TCSVT.2021.3072207

Publication date:
2022

Citation for published version (APA):
Zhang, T., Liu, X., Zhang, Q., & Han, J. (2022). SiamCDA: Complementarity-and distractor-aware RGB-T tracking based on Siamese network. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(3), 1403-1417. https://doi.org/10.1109/TCSVT.2021.3072207

SiamCDA: Complementarity- and distractor-aware RGB-T tracking based on Siamese network

Tianlu Zhang, Xueru Liu, Qiang Zhang* and Jungong Han*

***Abstract*—Recent years have witnessed the prevalence of using the Siamese network for RGB-T tracking because of its remarkable success in RGB object tracking. Despite their faster than real-time speeds, existing RGB-T Siamese trackers suffer from low accuracy and poor robustness, compared to other state-of-the-art RGB-T trackers. To address such issues, a new complementarity- and distractor-aware RGB-T tracker based on Siamese network (referred to as SiamCDA) is developed in this paper. To this end, several modules are presented, where the feature pyramid network (FPN) is incorporated into the Siamese network to capture the cross-level information within unimodal features extracted from the RGB or the thermal images. Next, a complementarity-aware multi-modal feature fusion module (CAMF) is specially designed to capture the cross-modal information between RGB features and thermal features. In the final bounding box selection phase, a distractor-aware region proposal selection module (DAS) further enhances the robustness of our tracker. On top of the technical modules, we also build a large-scale, diverse synthetic RGB-T tracking dataset, containing more than 4831 pairs of synthetic RGB-T videos and 12K synthetic RGB-T images. Extensive experiments on three RGB-T tracking benchmark datasets demonstrate the outstanding performance of our proposed tracker with a tracking speed over 37 frames per second (FPS).**

***Index Terms*—RGB-T tracking, Siamese network, Complementarity-aware fusion, Distractor-aware region proposal selection, Large-scale synthetic dataset**

I. Introduction

VISUAL object tracking aims to estimate the position of an arbitrary target in a video sequence, given only its location in the first frame. It is a fundamental research task in computer vision, and facilitates numerous practical applications such as visual surveillance [1], unmanned vehicles [2] and human-computer interactions [3]. With the exploration of mathematical modeling techniques, especially deep neural networks, recently advanced tracking approaches focus on employing large labeled video datasets to train an end-to-end network in an offline way. Their performance has witnessed a continuous improvement with the aid of more effective network architectures and more publicly available tracking datasets. Despite remarkable advances, most visual tracking algorithms focus on unimodal tracking, especially on RGB tracking, which remains challenging due to many factors such as heavy occlusion, large deformation, and illumination variations.

(a) (b)

Fig. 1. Illustration of the complementary information between multi-modal images. (a) and (b) are two examplar frames from RGB modality and thermal modality, respectively. As shown in the regions marked by the red boxes, obvious complementary information exists between the two modality images, which will benefit the subsequent tracking task.

Recently, some researchers attempt to apply multi-modal data, such as RGB-thermal (RGB-T) images [4] and RGB-depth (RGB-D) images [5], to improve the performance of trackers. Among them, RGB-T tracking has attracted more attention because of the complementarity between RGB images and thermal images. RGB images can capture rich target information but are susceptible to environments. Thermal images are not sensitive to illumination change and have strong ability to penetrate the haze but they lack the detailed texture information of the targets [6], [7]. As shown in Fig. 1, RGB and thermal images can be applied together to provide complementary information for object tracking. In light of it, we focus on RGB-T tracking in this paper, to address some issues arising from unimodal RGB tracking.

So far, many RGB-T trackers have been put forward. Early works [8]–[10] are based on manually extracted features. Generally, these methods cannot well adapt to challenging environments, such as drastic appearance changes, clutter backgrounds, rapid movements of targets and occlusion. Inspired by the success of Convolution Neural Networks (CNNs) in RGB tracking, there are several attempts [11]–[13], using CNNs to improve the performance of RGB-T trackers. Owing to the robust feature extraction and representation ability of deep CNNs, these newly developed RGB-T trackers usually outperform those traditional ones by a clear margin. Therefore,

Tianlu Zhang, Xueru Liu and Qiang Zhang are with Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an, Shaanxi 710071, China and also with Center for Complex Systems, School of Mechano-Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071, China. Email: tianluzhang@stu.xidian.edu.cn, xueruliu@stu.xidian.edu.cn and qzhang@xidian.edu.cn.

Jungong Han is with Computer Science Department, Aberystwyth University, SY233FL, UK. Email: jungonghan77@gmail.com.

*Corresponding authors: Qiang Zhang and Jungong Han.

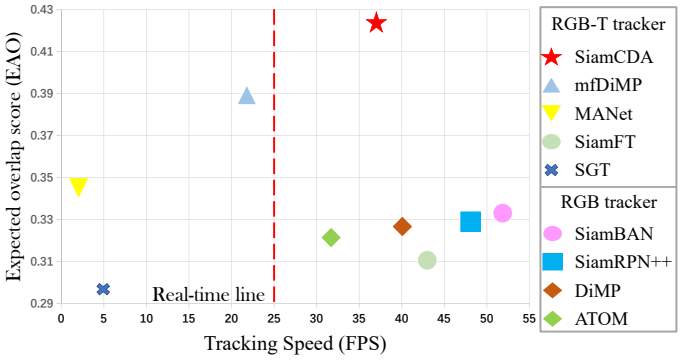

Fig. 2. Comparison of performance and speed for some state-of-the-art tracking methods on VOT-RGBT2019 [15]. We visualize the Expected Average Overlap (EAO) with respect to the Frames-Per Second (FPS). Closer to the top means higher precision, and closer to right means faster. SiamCDA is able to rank the 1st in EAO while running at 37 FPS.

some trackers [12], [13] based on Multi-Domain Network [14] appeared in recent years. However, such trackers have to be put on the shelf because of their far slower than the real-time speeds.

To remedy this situation, inspired by the success of Siamese network in unimodal object tracking, recent research has shifted to apply the Siamese network [16] to speed up the RGB-T tracking algorithms. Zhang et al. [17] took the first attempt in applying the fully convolutional Siamese network to RGB-T tracking. Their experiments demonstrate that the running speed of the proposed RGB-T Siamese tracker, dubbed SiamFT, can reach around 30 FPS, thus meeting the real-time requirement. Although they have achieved faster speed, there continues to be a huge gap in performance compared to most state-of-the-art RGB-T tracker, as shown in Fig. 2.

In this paper, we design a unified RGB-T tracking framework based on Siamese network, referred to as SiamCDA, which can achieve high performance, while maintaining real-time running speed. As shown in Fig. 2, our tracker can achieve equivalent performance to some state-of-the-art RGB-T trackers with much more effective tracking speed. The proposed multi-modal Siamese tracking algorithm carries out the following four steps:

(1) Siamese network for unimodal feature extraction Most of the existing RGB-T trackers have built their networks upon tailored AlexNet [18] or VGGNet [19], which usually contains no more than five convolutional layers. Deeper neural networks, such as ResNet [20], ResNeXt [21] and MobileNet [22], have been proven to be effective in unimodal trackers based on Siamese network by virtue of a spatial-aware sampling strategy [23]. Motivated by that, in our proposed method, we also adopt the modified ResNet-50 [20] as our backbone network for the unimodal RGB and thermal image feature extraction. As well, different levels of the extracted unimodal features usually contain varieties of spatial or semantic information for tracking. Taking full advantage of the low-level and high-level features may improve the tracking accuracy to some extent. Considering that, we will append several feature pyramid networks (FPNs) [24] on the backbone network to capture the cross-level features within the extracted unimodal RGB and thermal features.

(2) Multi-modal feature fusion

For the RGB-T tracking task, how to effectively fuse the RGB and thermal information is one of the most important issues. Several methods have been proposed, such as element-wise summation [25], concatenation [26] and content-dependency weighting based fusion strategies [12] [17]. However, most of these existing fusion strategies do not consider the feature differences between the input RGB and thermal images during fusion. In fact, RGB images and thermal images have different imaging mechanisms and their features have large differences (e.g., polarity reverse). Directly fusing the original or weighted unimodal RGB and thermal features will reduce the discriminability of the fused features, thus degrading the subsequent tracking performance. In order to fully exploit the cross-modal features within the input RGB and thermal images, a complementarity-aware multi-modal feature fusion module (CA-MF) will be designed in our tracker. In CA-MF, the differences between unimodal RGB features and thermal features are first reduced by introducing complementary (or additional) information from one modality data to the other modality data. Then the enhanced RGB and thermal features are further combined to obtain the final fused features via some fusion strategies (e.g., concatenation).

(3) Siamese region proposal

Similar to the RGB Siamese trackers, RGB-T Siamese trackers also formulate the tracking problem as the cross-correlation between the fused multi-modal features of template images and those of detection images. In addition, as discussed in [23] and [27], the application of regional proposal network (RPN) may make the Siamese trackers have a great advantage in generating accurate bounding boxes. Subsequently, we will also apply several Siamese RPNs [23] in our proposed RGB-T tracker to promote the tracking accuracy.

(4) Region proposal selection

The outputs of the Siamese RPNs are a set of bounding boxes with their corresponding confidence scores. In order to get the final tracking box, Siamese trackers generally apply cosine window and scale change penalty [27] to re-rank the proposals' score. These may work well in most cases. However, in some special cases, there may exist some semantic backgrounds, such as objects having similar attributes with targets, which are usually considered as distractors. Because of being trained completely offline, these Siamese trackers are generally not discriminative enough to handle such distractors, thus easily leading to tracking drift. To solve this issue, a distractor-aware region proposal selection module (DAS) will be specially designed in this paper to further improve the robustness of our proposed tracker against the distractors. In DAS, whether there exist distractors in the current tracking frame is first determined, according to which, the final target box is further selected.

In addition, the training data has significant effects on the performance of a Siamese tracker. So far, some RGB-T tracking datasets have appeared, such as VOT-RGBT2019 [15], RGBT234 [28] and GTOT [9]. Among these datasets, the RGBT234 dataset [28] is the largest one. In spite of that, RGBT234 just contains 234 pairs of RGB and thermal annotated videos. This is far from enough to train a Siamese

tracker, thus limiting the tracking performance. For that, a large-scale synthetic RGB-T tracking dataset, containing more than 4831 synthetic RGB-T videos and 12K RGB-T images, will be first constructed. Then these synthetic RGB-T videos (or images) and several real RGB-T datasets will be simultaneously employed to train our tracker. This will significantly enhance the feature representation ability of the tracking model and further improve the tracking performance of our proposed tracker.

In summary, this work has the following four-fold main contributions:

- A unified RGB-T tracking framework based on Siamese network is designed to achieve high tracking performance but still maintain the tracking efficiency by introducing several modules.
- A complementarity-aware multi-modal feature fusion module (CA-MF) is presented to enhance the discriminability of the fused features by first reducing the modality-differences between unimodal features and then fusing them. A distractor-aware region proposal selection module (DAS) is proposed to improve the tracker's robustness by first determining the distractors in each frame and then selecting the final bounding box for tracking. These two proposed modules significantly improved the performance of RGB-T Siamese tracker.
- A large-scale synthetic RGB-T tracking dataset is built, which contains more than 4831 pairs of synthetic RGB-T videos and 12K synthetic RGB-T images. The newly constructed synthetic RGB-T dataset and several existing real RGB-T datasets are jointly employed to train our model for further improving the tracking performance.
- Our tracker achieves new state-of-the-art performance on VOT-RGBT2019 [15], RGBT234 [28] and GTOT [9] with the speed of 37 FPS.

II. Related Work

A. RGB Tracking methods

In the last few years, visual object tracking has made rapid progress because of the presence of some new benchmark datasets and advanced methodologies. Especially, with the great success of CNNs in various computer vision tasks, numerous trackers based on deep feature representations have emerged [29]–[32] and obtained new state-of-the-art tracking performance in popular tracking benchmarks.

These modern tracking algorithms or models can be roughly categorized into two branches: discriminative trackers and generative ones. Discriminative trackers train a classifier to distinguish the target from the background, which ordinarily demands to train models online. For example, in [14], a new CNN architecture, referred to as Multi-Domain Network (MDNet), was presented to learn the shared representation of targets from multiple annotated video sequences for tracking. Recently, some newly developed trackers, such as ATOM [31] and DiMP [33], have also been presented to achieve more outstanding performance. Usually, these discriminative trackers achieve very promising tracking results while having relatively slow speed.

Differently, generative trackers [16] [27] [34] find the candidates that match the targets the best by computing their joint probability densities between targets and search candidates. Particularly, as a typical kind of generative trackers, the Siamese network based trackers [16] [27] have received surging attention in that their performance have taken the lead in various benchmarks while running at real-time speed.

B. Siamese network based RGB trackers

As one of the pioneering works, SiamFC [16] constructed a fully convolutional Siamese network to train a tracker. The key idea of SiamFC was to formulate the object tracking task as a similarity learning problem. In order to achieve more accurate target bounding boxes, SiamRPN [27] introduced the region proposal network [35] into SiamFC. Encouraged by the success of SiamFC and SiamRPN, many researchers [36]–[38] follow these works and presented some improved models. Zhu et al. [36] extended the SiamRPN by developing distractor-aware training. C-RPN [37] proposed a multi-stage tracking framework to make localization more accurate. However, the performance of the Siamese network based trackers can not move on with deeper network as the backbone. Aiming to solve this problem, SiamRPN++ [23] employed a new strategy during model training, i.e., randomly shifting the training object location in the search region, and introduced modern deep neural networks, such as ResNet [20], ResNeXt [21] and MobileNet [22], into the Siamese network based trackers. SiamDW [39] designed a residual network for visual tracking with controlled receptive field size and network stride. Owing to these modifications, better tracking accuracy can be achieved by using a very deep network architecture. Recently, inspired by some anchor-free detectors, such as [40] and [41], several Siamese trackers [42]–[44] regressed the distance from the estimated target center to its sides of a bounding box. At the same time, some researchers [45] tried to use adversarial attack of CNN to improve the robustness of deep learning trackers.

C. RGB-T Tracking methods

In recent years, some RGB-T tracking algorithms [8], [12], [13] have been presented to exploit RGB and thermal images to boost tracking performance. The early RGB-T tracking algorithms [8]–[10] are based on some handcrafted features. With the development of deep learning, more and more RGB-T trackers based on deep features [12], [13], [26] are presented. These RGB-T trackers are usually designed on the basis of RGB trackers. For example, in [12] and [13], MDNet was used as their baseline trackers. Particularly, in [12], Zhu et al. first proposed a network to aggregate features of all layers and all modalities, and then pruned these features to reduce noise and redundancies. In [13], Li et al. proposed a multi-adapter architecture to learn modality-shared, modality-specific, and instance-aware target representations, respectively. In addition, Zhang et al. [26] introduced DiMP [33] as their baseline tracker and investigated different levels of fusion mechanisms to find the optimal fusion architecture. Their experimental

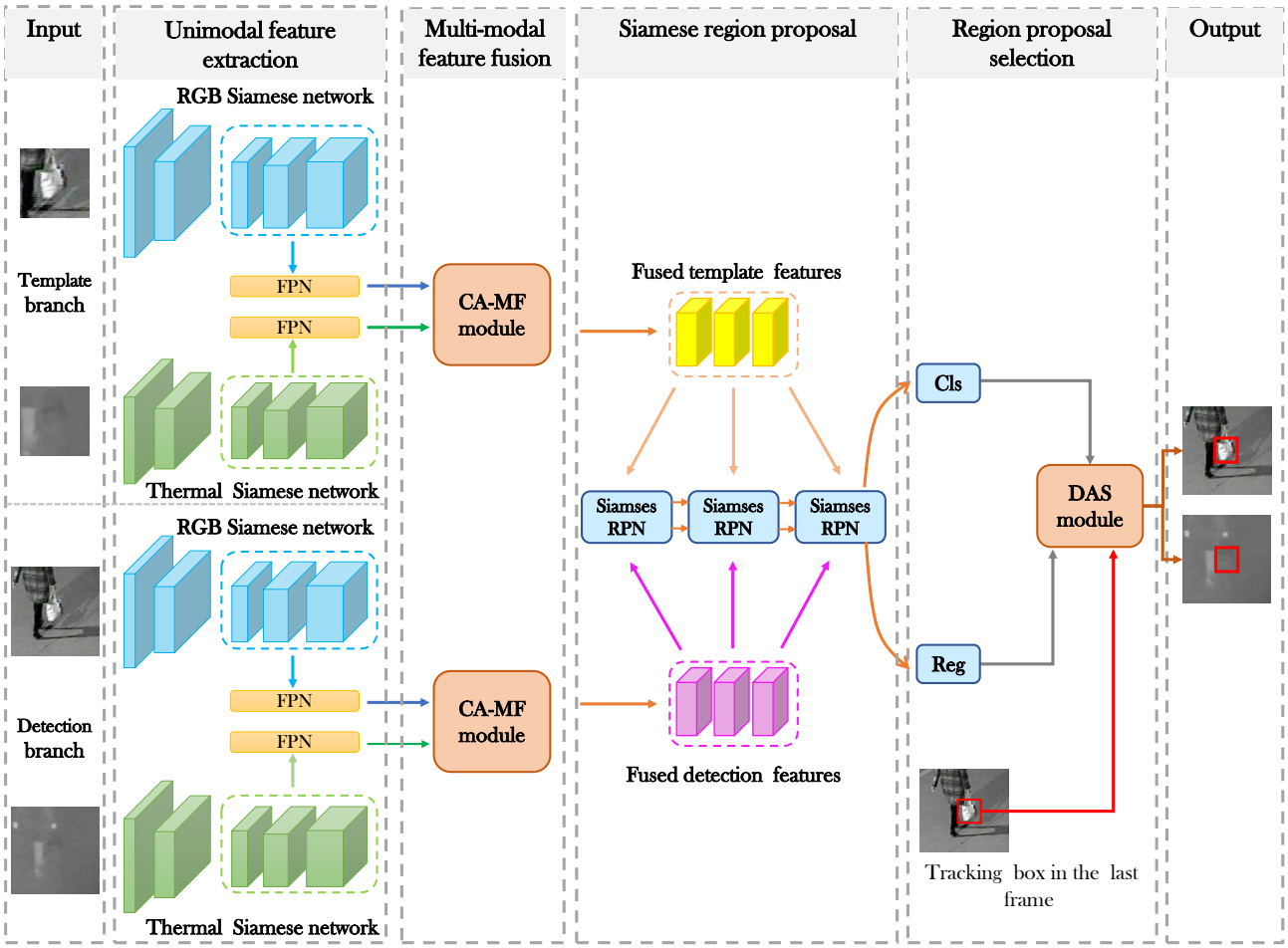

Fig. 3. An overview of the proposed SiamCDA. The overall network consists of four main parts: Siamese network for unimodal feature extraction, CA-MF module for multi-modal feature fusion, SiameseRPNs for region proposal generation and DAS module for region proposal selection.

results demonstrated that their proposed fusion tracker significantly improved the performance of the baseline tracker with respect to unimodal tracking and achieved new state-of-the-art results. However, these discriminative RGB-T trackers achieve high tracking performance at a cost of huge computational complexity. For example, the tracking speed of the MANet [13] is only about 2 FPS.

Considering the successful application of Siamese network in RGB tracking, some works also try to introduce the Siamese network to RGB-T tracking to improve computational efficiency. For instance, SiamFT [17] applied two Siamese networks to extract the unimodal features from the input RGB and thermal images, respectively, and used some hand-designed modality weights to calculate the weights of different modalities. DSiamMFT [25] designed an RGB-T tracker based on dynamic Siamese network [46] with multi-layer fusion. However, these works are the initial exploration of applying Siamese network in RGB-T tracking. In addition, the lack of large scale RGB-T training dataset also hinders the performance of these trackers. As a result, there is still a large gap in tracking precision between these Siamese network based RGB-T trackers and other state-of-the arts, although they may achieve real-time tracking speeds.

III. Method

In this section, we describe our proposed RGB-T tracking model in detail. As shown in Fig. 3, the overall model consists of four parts: Siamese networks for unimodal feature extraction, CA-MF modules for multi-modal feature fusion, region proposal networks for proposal generation and DAS module for region proposal selection. In the subsequent subsections, we will discuss each part in details.

A. Siamese networks for unimodal feature extraction

In our proposed tracking model, two Siamese networks, namely RGB Siamese network and Thermal Siamese network, are employed to extract the unimodal features from the RGB and thermal images, respectively. The two Siamese networks share the same structure but different parameters to well extract the unimodal features from each input image. Each Siamese network further consists of two branches that share the same structures and parameters. One branch (called the template branch) is used to extract the features from template images. The other branch (called the detection branch) is used to extract the features from search images. Moreover, let φ_{rgb} denote one of the branches (i.e., the template branch or detection branch) in the RGB Siamese network. x_{rgb} and z_{rgb} denote the template patch and the search patch for the inputs

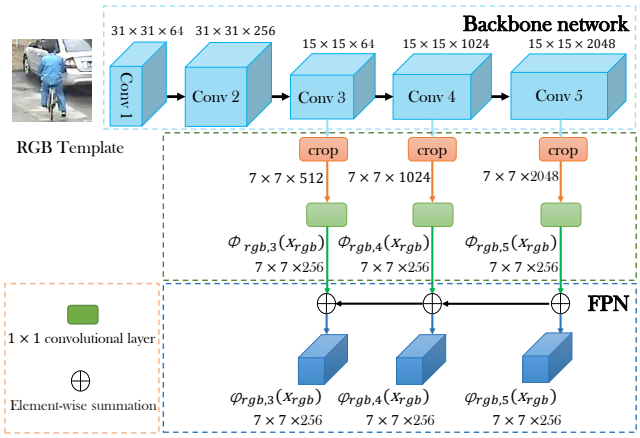

Fig. 4. Illustration of the template branch of RGB Siamese network. It consists of a backbone network and a feature pyramid network (FPN).

of RGB Siamese network, respectively. Similarly, φ_t denotes one of the branches in the thermal Siamese network. x_t and z_t denote the template patch and the search patch for the corresponding inputs of thermal Siamese network, respectively. The output features of the two-stream Siamese networks are thus represented as $\varphi_{rgb}(x_{rgb})$, $\varphi_{rgb}(z_{rgb})$, $\varphi_t(x_t)$ and $\varphi_t(z_t)$, respectively.

In the following contents, we will take the template branch in the RGB Siamese network as an example to discuss the construction of our unimodal feature extraction module, considering that all of the four branches in the two-stream Siamese networks share the same structure. As shown in Fig. 4, the RGB template feature extraction branch consists of a backbone network and a feature pyramid network (FPN) [24].

Motivated by [23], we also adopt ResNet50 [20] as our backbone network, considering that deeper networks may significantly boost the performance of the Siamese trackers. As well, to increase the spatial resolution of the feature maps and preserve more detailed information about the template patch (or detection patch), we remove the down-sampling operations and set the stride to 1 in the last two convolutional blocks (i.e., $conv4$ and $conv5$ blocks). We also employ the atrous convolution with different atrous rates, instead of the traditional convolution, in the last two blocks to increase the receptive fields without decreasing the spatial resolutions of the feature maps. Specifically, the atrous rates in $conv4$ and $conv5$ blocks are set to 2 and 4, respectively. An 1×1 convolutional layer is further appended to each of the last three blocks (i.e., $conv3$, $conv4$ and $conv5$ blocks) to reduce the channels of the output feature maps to 256. Finally, in order to reduce the computational burden of our proposed model, only the features from the center 7×7 regions are selected as the template features. Thus, we may obtain three levels of the template features (denoted by $\phi_{rgb,3}(x_{rgb})$, $\phi_{rgb,4}(x_{rgb})$ and $\phi_{rgb,5}(x_{rgb})$, respectively) with the same sizes from the last three blocks of the backbone network, which will be used in the subsequent tracking task. Here, the features from the first two blocks (i.e., $conv1$ and $conv2$) are not used considering that they may contain much more disturbing information for tracking.

The three levels features extracted from the backbone network contain different information about the template (or detection) patch. The low-level features (e.g., $\phi_{rgb,3}(x_{rgb})$) may contain more visual attributes, like edges, corners, colors and shapes, which are indispensable for the location of the objects, while the high-level features (e.g., $\phi_{rgb,5}(x_{rgb})$) may contain more semantic attributes that are crucial for the discrimination of the objects. Therefore, the fusion of the low-level and high-level features will improve the tracking accuracy to some extent.

Considering that, we append a feature pyramid network (FPN) on the last three blocks of our backbone network to capture the cross-level features within the three levels features $\{\phi_{rgb,i}(x_{rgb}), (i = 3, 4, 5)\}$. Owing to the fact that the three levels of features $\{\phi_{rgb,i}(x_{rgb}), (i = 3, 4, 5)\}$ extracted from the backbone network have the same spatial resolutions and the same number of channels, the output features of FPN for each level $\{\varphi_{rgb,i}(x_{rgb}), (i = 3, 4, 5)\}$ can be obtained by a top-down integration way, i.e.,

$$\varphi_{rgb,5}(x_{rgb}) = \phi_{rgb,5}(x_{rgb}), \tag{1}$$

$$\varphi_{rgb,4}(x_{rgb}) = \phi_{rgb,4}(x_{rgb}) \oplus \varphi_{rgb,5}(x_{rgb}), \tag{2}$$

$$\varphi_{rgb,3}(x_{rgb}) = \phi_{rgb,3}(x_{rgb}) \oplus \varphi_{rgb,4}(x_{rgb}), \tag{3}$$

where $\oplus$ denotes the element-wise addition. The outputs of the FPNs are also seen as the output features of the template branch in the RGB Siamese network, i.e., $\varphi_{rgb}(x_{rgb}) = \{\varphi_{rgb,i}(x_{rgb}), (i = 3, 4, 5)\}$. The output features of the detection branch in the RGB Siamese network $\varphi_{rgb}(z_{rgb}) = \{\varphi_{rgb,i}(z_{rgb}), (i = 3, 4, 5)\}$, the output features of the template and detection branches in the thermal Siamese network $\varphi_t(x_t) = \{\varphi_{t,i}(x_t), (i = 3, 4, 5)\}$ and $\varphi_t(z_t) = \{\varphi_{t,i}(z_t), (i = 3, 4, 5)\}$ are obtained in the similar way.

B. CA-MF modules for multi-modal feature fusion

Given the features from the RGB and thermal Siamese networks, the next step is to obtain the fused template features and the fused detection features for tracking. As that in the existing RGB-T Siamese trackers [17] [25], the features from the template branch in the RGB Siamese network and the corresponding template features from the thermal Siamese network are combined to obtain the fused template features for tracking. Similarly, the RGB detection features and their corresponding thermal detection features are combined to obtain the fused detection features for tracking. Then, how to fuse them to effectively capture the cross-modal complementary information between the RGB and thermal images is an especially important issue in RGB-T tracking models.

The most straightforward and commonly used ways for the fusion of multi-modal features are element-wise summation [25] and concatenation [26]. However, these fusion strategies often ignore the feature reliability from each modality and cannot effectively leverage the cross-modal complementary information within the multi-modal RGB and thermal images. In [12], a content-dependency weighting based fusion strategy was presented to fuse the multi-modal RGB and thermal features for tracking. Owing to the consideration of

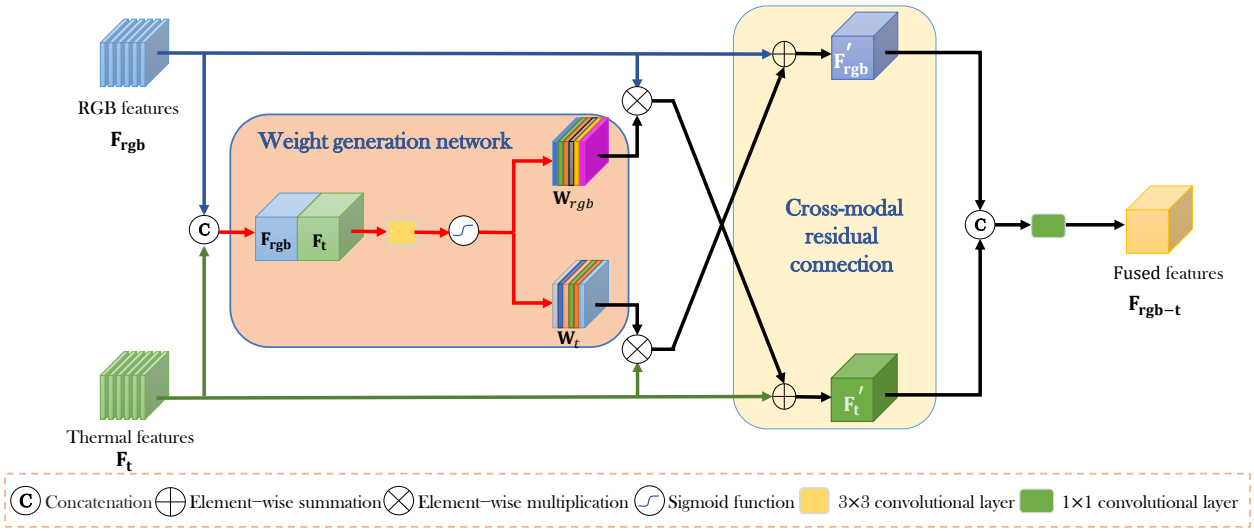

Fig. 5. Illustration of our proposed CA-MF module. First, the weight generation sub-network takes the features maps from two-stream Siamese networks as input and produces weights that reflect how much additional information should be introduced from one modality data to another modality data. Then the enhanced unimodal features are obtained by using cross-modal residual connections. Finally, the fused features are obtained by performing some concatenation and convolutional operations on these enhanced unimodal features.

the feature reliability of each modality data, the content-dependency weighting based fusion strategy usually achieves better performance than those simple element-wise summation or concatenation based ones. Despite that, most of these existing fusion strategies do not consider the feature differences between the input multi-modal RGB and thermal images during fusion.

Alternatively, if the differences between RGB features and thermal features are first reduced to some extent before they are fed into the fusion module, more complementary information between the input multi-modal images will be preserved into the fused features and will improve the discriminability of fused features, thus further benefiting the subsequent tracking task. For that, a complementarity-aware multi-modal feature fusion module (CA-MF) is presented in this paper. In CA-MF, the unimodal features from one modality data (e.g, RGB images or thermal images) will be first reinforced by introducing additional information from the other modality data. Doing so potentially reduces the difference between the unimodal RGB and thermal features. Afterwards, the enhanced RGB and thermal features will be further combined to achieve the final fused features via some fusion strategies.

As shown in Fig. 5, the proposed CA-MF consists of a weight generation sub-network and two cross-modal residual connections. The weight generation sub-network concatenates the unimodal feature maps $\mathbf{F}_{rgb}$ and $\mathbf{F}_t$ as the inputs and outputs two weight maps $\mathbf{W}_{rgb}$ and $\mathbf{W}_t$ of the same spatial size as those of $\mathbf{F}_{rgb}$ (or $\mathbf{F}_t$) via a convolutional layer and a sigmoid layer. Specifically, the convolutional layer has one filter with kernel size 3×3 and produces two weight maps. The subsequent sigmoid layer is used to normalize the values of the two weight maps to [0,1]. Mathematically, the weight generation sub-network is expressed by

$$\mathbf{W}_{rgb}, \mathbf{W}_t = \sigma(conv(cat(\mathbf{F}_{rgb}, \mathbf{F}_t), \theta_1)), \tag{4}$$

where $cat(*)$ denotes the concatenation operation and $conv(*, \theta_1)$ denotes the convolutional layer with parameters θ_1. $\sigma(*)$ denotes the sigmoid layer. It should be also noted that the weights $\mathbf{W}_{rgb}$ and $\mathbf{W}_t$, generated by using Eq.4, reflect how much the additional information should be introduced from one modality data to another modality data, instead of the importance or reliability of each modality data as in [12].

With the weights $\mathbf{W}_{rgb}$ and $\mathbf{W}_t$, the enhanced unimodal features $\mathbf{F}'_{rgb}$ and $\mathbf{F}'_t$ are then obtained by using cross-modal residual connections, i.e.,

$$\mathbf{F}'_{rgb} = \mathbf{F}_{rgb} + \mathbf{F}_t \otimes \mathbf{W}_t, \tag{5}$$

$$\mathbf{F}'_t = \mathbf{F}_t + \mathbf{F}_{rgb} \otimes \mathbf{W}_{rgb}, \tag{6}$$

where $\otimes$ denotes the element-wise multiplication. As shown in Eq. 5, the enhanced unimodal RGB features $\mathbf{F}'_{rgb}$ contain some additional thermal features in addition to the original unimodal RGB features $\mathbf{F}_{rgb}$. Similarly, as shown in Eq. 6, the enhanced unimodal thermal features $\mathbf{F}'_t$ contain some additional RGB features as well as the original unimodal thermal features $\mathbf{F}_t$. The differences between the original multi-modal features $\mathbf{F}_{rgb}$ and $\mathbf{F}_t$ will be reduced to some extent by using the cross-modal residual connections.

The finally fused features $\mathbf{F}_{rgb-t}$ are thus obtained by further performing some concatenation and convolutional operations on these enhanced unimodal features $\mathbf{F}'_{rgb}$ and $\mathbf{F}'_t$, i.e.,

$$\mathbf{F}_{rgb-t} = conv(cat(\mathbf{F}'_{rgb}, \mathbf{F}'_t), \theta_2). \tag{7}$$

Here, $conv(*, \theta_2)$ denotes the convolutional layer with kernel size 1×1 and parameters θ_2. As shown in Eq. 7, the enhanced unimodal features, instead of the original unimodal features, are used to obtain the final fused features in our proposed CA-MF. This will improve the discriminability of the fused features for the subsequent RGB-T tracking.

Given the three levels of template features $\varphi_{\text{rgb}}(x_{\text{rgb}}) = \{\varphi_{\text{rgb,i}}(x_{\text{rgb}}) \mid i = 3, 4, 5\}$ from the RGB Siamese network and the three levels of template features $\varphi_t(x_t) =$

$\{\varphi_{t,i}(x_t) \mid i=3,4,5\}$ from the thermal Siamese network, three levels of fused template features $\mathbf{F}^T_{rgb-t} = \left\{\mathbf{F}^T_{rgb-t,i} \mid i=3,4,5\right\}$ are obtained by performing the proposed CA-MFs on each level of RGB and thermal template features, respectively. Similarly, three levels of fused detection features $\mathbf{F}^D_{rgb-t} = \left\{\mathbf{F}^D_{rgb-t,i} \mid i=3,4,5\right\}$ are also obtained by performing several CA-MFs on the RGB detection features $\varphi_{\text{rgb}}(z_{\text{rgb}}) = \{\varphi_{\text{rgb},i}(z_{rgb}) \mid i=3,4,5\}$ and the thermal detection features $\varphi_t(z_t) = \{\varphi_{t,i}(z_t) \mid i=3,4,5\}$ with the same levels, respectively.

C. Siamese region proposal networks for proposal generation

Similar to RGB Siamese trackers, we apply three Siamese regional proposal networks (RPNs) on the fused template features and the fused detection features. Each Siamese RPN is employed for one level of the fused template and detection features and generates a set of regional proposals individually. The outputs of the three Siamese RPNs are further combined via a weighted fusion layer to achieve the final outputs of the Siamese RPNs.

Specifically, each Siamese RPN has two branches, one for foreground-background classification and the other for proposal regression. If there are k anchors, the network outputs $2k$ channels for classification and $4k$ channels for regression. Furthermore, as described in [23], each branch in the Siamese RPN consists of two fully convolutional layers and a depth-wise cross-correlation layer with a classification (or regression) head on top.

More specifically, for the $i-th(i=3,4,5)$ level, the fused template features $\mathbf{F}^T_{rgb-t,i}$ and the fused detection features $\mathbf{F}^D_{rgb-t,i}$ are first, respectively, fed into the two fully convolutional layers in the classification branch of the Siamese RPN to modify their characteristics for classification, thus obtaining the modified template features $\tilde{\mathbf{F}}^T_{rgb-t,i}$ and detection features $\tilde{\mathbf{F}}^D_{rgb-t,i}$. These modified features are further fed into the depth-wise cross-correlation layer and obtain the correlation features $\mathbf{F}^{cls}_i$. Finally, the correlation features $\mathbf{F}^{cls}_i$ are fed into a convolutional layer that having $2k$ filters with kernel size 1×1 and a softmax layer to achieve the final classification score maps $\mathbf{S}^{cls}_i \in R^{w\times h\times 2k}$. Here, $w\times h$ denotes the spatial sizes of the score map in each channel of $\mathbf{S}^{cls}_i$. Similarly, a set of regression maps $\mathbf{S}^{reg}_i \in R^{w\times h\times 4k}(i=3,4,5)$ are obtained by feeding the fused template features $\mathbf{F}^T_{rgb-t,i}$ and the fused detection features $\mathbf{F}^D_{rgb-t,i}$ into the regression branch of the Siamese RPN. The final classification score maps $\mathbf{S}^{cls} \in R^{w\times h\times 2k}$ and regression maps $\mathbf{S}^{reg} \in R^{w\times h\times 4k}$ are thus obtained by using a weighted summation way via a fusion layer as in [23]. Each spatial position on $\mathbf{S}^{cls}$, containing a $2k$ vector, provides the negative and positive activation of each anchor at the corresponding location on the original map. Correspondingly, each spatial position on $\mathbf{S}^{reg}$, containing a $4k$ vector, measures the distance between each anchor and its corresponding ground truth.

D. DAS module for region proposal selection

Based on the proposal classification maps $\mathbf{S}^{cls}$ and bounding box regression maps $\mathbf{S}^{reg}$, outputs from the RPNs

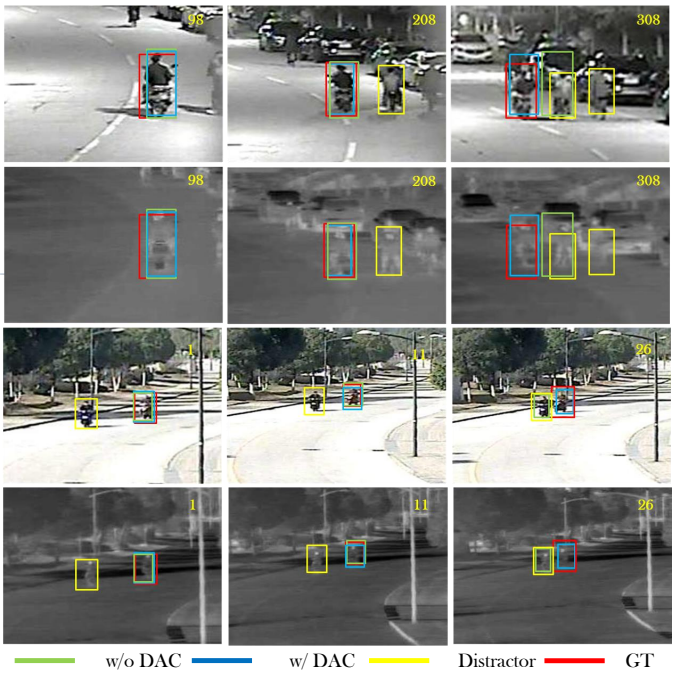

Fig. 6. Illustrations of the validity of the proposed DAS module. As shown in the figure, the proposed DAS module can greatly reduce the tracking drifts that are caused by distractors.

mentioned above, $(w\times h\times k)$ bounding boxes $\mathbf{B} = \left\{B_i = \left(x^{pro}_i, y^{pro}_i, w^{pro}_i, h^{pro}_i, s^{cls}_i\right)^T \mid i\in(w\times h\times k)\right\}$ together with their corresponding confidence scores are thus obtained as in [27]. Here, $\left(x^{pro}_i, y^{pro}_i\right)^T$ denotes the center point of the $i-th$ bounding box B_i. w^{pro}_i and h^{pro}_i are its width and height, respectively. s^{cls}_i denotes the classification score of B_i.

In this subsection, we will discuss how to select the final bounding box from $\mathbf{B}$ for the tacking task. Due to the fully offline training strategy, Siamese trackers are not discriminative enough to distinguish two objects with similar attributes, as shown in Fig. 6. These semantic backgrounds are usually considered as distractors. Existing trackers based on Siamese network, including RGB trackers and RGB-T trackers, usually first adopt the cosine window and scale change penalty [27] to re-rank the proposal's confidences scores and then select the bounding box with the highest confidence score as the final tracking one. Although they could suppress large displacements, these existing post-processing methods cannot effectively suppress the interference and are prone to tracking drift when distractors are close to the target.

To improve the robustness of our tracker, we will design a distractor-aware selection module (DAS) in this subsection. Unlike the existing post-processing methods, which only consider the re-ranked confidence scores of each bounding box, the proposed DAS module not only considers the confidence score of each bounding box, but also considers the influences of distractors on object tracking as well as the continuity of target motion between consecutive frames. The proposed module is based on the following two characteristics in the tracking task: (1) Distractors are usually accompanied by high confidence scores; (2) The displacement of the target between

two adjacent frames is usually not very far away. Specific steps of DAS are thus as follows:

(1) Determine the distractors from the bounding box set $\mathbf{B}$. More specifically, the bounding box whose confidence score is smaller than a threshold v_e is first removed from the bounding box set $\mathbf{B}$, i.e.,

$$B_i = \begin{cases} preserved & if\ s_i^{cls} > v_e \\ removed & if\ s_i^{cls} < v_e \end{cases}. \tag{8}$$

Here, the threshold v_e is experimentally set to 0.3. Then the Non-maximum Suppression (NMS) is performed on those preserved bounding boxes to further remove those bounding boxes with smaller confidence scores, thus obtaining a new bounding box set $\mathbf{B}^{'}$. In addition, those bounding boxes with large scale or ration changes, i.e., those bounding boxes that do not satisfy Eq.9 and Eq.10, will also be removed from $\mathbf{B}^{'}$, considering that these bounding boxes could not be distractors.

$$(1 - v_a) * size[B_l^*] < size[B_i] < (1 + v_a) * size[B_l^*], \tag{9}$$

$$(1 - v_r) * ratio[B_l^*] < ratio[B_i] < (1 + v_r) * ratio[B_l^*], \tag{10}$$

where $size[\cdot]$ denotes the area of a bounding box and $ratio[\cdot]$ denotes the ratio of the height of a bounding box to its width. v_a and v_r are both experimentally set to 0.2. B_l^* denotes the tracking bounding box determined in the last frame. B_i denotes a bounding box in $\mathbf{B}^{'}$. After that, another new bounding box set $\mathbf{B}''$ that containing K bounding boxes is obtained. In $\mathbf{B}''$, only one bounding box may be the target and the other $K-1$ bounding boxes will be seen as the distractors.

(2) Determine the initial candidate box B_{init} from the original bounding box set $\mathbf{B}$. For that, the confidence scores s_i^{cls} of each bounding box B_i in $\mathbf{B}$ is multiplied by a penalization term p_i to suppress large changes in size and ratio, i.e.,

$$p_i = e^{v_p \times \max\left(\frac{r_i}{r_l^*}\right) \times \max\left(\frac{z_i}{z_l^*}\right)}, \tag{11}$$

$$\bar{s}_i^{cls} = s_i^{cls} \times p_i, \tag{12}$$

where $\bar{s}^{cls}$ denotes the new confidence score for the current bounding box. v_p is a hyper parameter that controls the magnitude of the penalization. r_i represents the ratio of the height of B_i to its width and r_l^* represents that of the tracking bounding box B_l^* in the last frame. z_i and z_l^* represents the areas of B_i and B_l^*, respectively. Then, a cosine window $\omega \in R^{w \times h \times k}$ with a window influence coefficient Ω is further performed on the new confidence scores $\bar{s}_i^{cls}$ to suppress the large displacement since smooth motion is assumed, i.e.,

$$\tilde{s}_i^{cls} = \bar{s}_i^{cls} \cdot (1 - \Omega) + \omega_i \cdot \Omega. \tag{13}$$

After that, the bounding box with the highest confidence score is selected as the initial candidate bounding box B_{init}.

(3) Determine the final tracking box B_c^* for the current frame. If there is no distractor, i.e., $K = 1$, in the first step, the initial candidate box B_{init} obtained in the second step is directly selected as the final tracking box B_c^* for the current frame. Differently, if there are some distractors in the current frame, i.e., $K > 1$, in the first step, according to the displacements of the target between two adjacent frames, two different selection strategies are further considered to select the final tracking box in our tracker. If the Intersection over Union (IoU) between B_{init} and the final tracking box B_l^* in the last frame is larger than a threshold v_o, the initial candidate box B_{init} obtained in the second step is still selected as the final tracking box B_c^* for the current frame. Otherwise, those bounding boxes whose IoU with the final tracking box B_l^* in the last frame is larger than a threshold v_s are first selected as the candidate boxes, and then the box with the highest confidence score s^{cls} in these candidates are selected as the final tracking box B_c^* for the current frame. Here, v_o and v_s are experimentally set to 0.2 and 0.7, respectively.

Despite its simplicity, DAS can improve the tracking robustness of our tracker to some extent. As shown in Fig. 6, our tracker with DAS can still accurately track the objects when distractors appear.

IV. RGB-T DATA GENERATION

The lack of large-scale labeled RGB-T tracking data will greatly reduce the performance of the RGB-T Siamese trackers, including our proposed tracker. Building a large scale RGB-T dataset is a time-consuming and energy-consuming task. Considering that, we will build a new large-scale synthetic RGB-T dataset (called LSS Dataset[1]) in this section, where some synthetic thermal images will be generated from real RGB images by using a newly proposed semantic-aware image-to-image translation method and some synthetic RGB videos will be generated from real thermal videos by using a video colorization method [47]. The newly built LSS dataset and several existing real RGB-T datasets will be simultaneously employed to train our proposed tracker for high tracking performance.

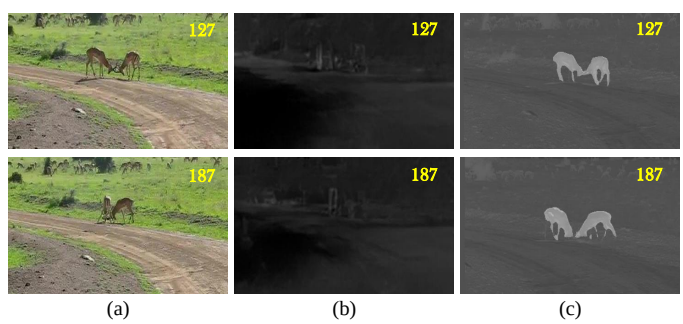

Fig. 7. Illustrations of some synthetic thermal images generated from RGB images by using different methods. (a) Original RGB images that are taken from the training set of VID [48]; (b) Generated thermal images by using pix2pix [49]; (c) Generated thermal images by using our proposed method.

A. Generate thermal images from RGB images

As in [26], the problem of generating thermal videos from RGB videos can be regarded as a problem of image-to-image translation. Many more image translation methods can be used to generate thermal images (or videos) from RGB videos in a frame-independent way. For example, in [26], pix2pix [49] was used to generate a large amount of thermal images from RGB images. However, most existing methods simply

[1]LSS dataset is available at https://github.com/RaymondCover/LSS-Dataset

treat the generation of thermal images from RGB images as an image style translation problem, without considering the imaging mechanism of thermal images. As a result, the generated thermal images could not well reflect the thermal infrared characteristics of an object in general. For example, as shown in Fig. 7, the objects are not obvious in some synthetic thermal images generated from the RGB videos. Obviously, by using such training data, the RGB-T tracker could not well learn how to fully exploit complementary information between RGB images and thermal images.

Thermal images are obtained by measuring the heats radiated from the objects. Different categories of objects usually show different gray values in thermal images. For example, people usually have higher gray values, while trees usually have lower gray values. Therefore, generating thermal images from RGB images may be regarded as a problem of setting different gray values for different categories of objects in the image. In addition, some existing RGB datasets provide more labeled information about the objects, such as category and location, which can also be used for image translation. Based on such information provided by these existing labeled RGB datasets and the imaging mechanism of thermal images, we propose a new semantics-aware image-to-image translation method to generate thermal images from RGB images, which can be divided into the following steps.

TABLE I
STATISTICAL AVERAGING VALUES AND VALUE RANGES FOR DIFFERENT CATEGORIES OF OBJECTS.

Category	Sub-categories in VID and COCO	Average value	Value range
animals and food	antelope; bear; bird; cattle; dog; domestic cat; elephant; fox; giant panda; hamster; horse; lion; lizard; monkey; rabbit; red panda; sheep; snake; squirrel; tiger; turtle; whale; zebra; person; food	141.46	110-175
vehicles	airplane; bicycle; bus; car; motorcycle; train; watercraft;	99.84	95-135
appliances	appliance; electronics	unknown	45-85
backgrounds	accessory; sports; outdoor objects; furniture;kitchenware; indoor objects	100.02	65-105

First, as shown in Fig. 8, we employ SiamMask [50] to convert the labels from bounding box level to pixel level, since the RGB tracking datasets usually annotate targets in the form of bounding boxes. Secondly, we set the gray value ranges of different objects according to their category information provided by the labeled dataset. Especially, we classify the targets in the VID dataset [48] and COCO dataset [51] into four categories, i.e., animals and food, vehicles, appliances, and backgrounds, as shown in Table I. Then we calculate the averaging gray values of different categories of targets in some other existing RGB-T datasets, such as RGBT234 [28] and GTOT [9], and experimentally set the gray value ranges of different categories of objects as shown in Table I. Lastly, for each object in the original RGB image, including the background area, its corresponding gray value range is assigned as follows, thus obtaining a synthesized thermal image:

(1) Get the initial gray value range for each object according to Table I. Specifically, for an RGB image to be transformed, we first get the pixel-level labels $\mathbf{MASK} = \{mask_i \mid i \in (1, N)\}$ of all objects in the image, including the background region. Here, $mask_i$ denotes the segmentation mask for the $i-th$ object in the image and N denotes the total number of objects (including background region) in the image. Then, according to Table I, we get the initial value range $(vmin_i, vmax_i)$ for the $i-th \quad (i = 1, 2, \ldots, N)$ object. For example, the initial gray value range for the object 'bull' in Fig. 8 is (110, 175).

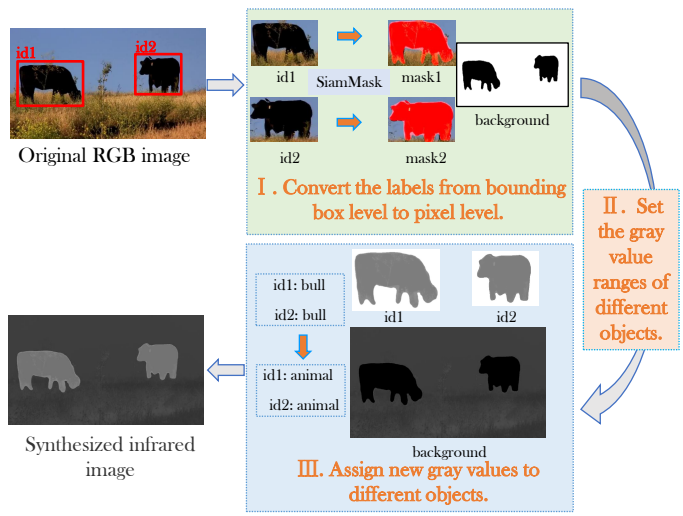

Fig. 8. Visualized process of the proposed semantics-aware image-to-image translation method.

(2) In order to improve the diversity of generated image scenes, we reset the gray value range from $(vmin_i, vmax_i)$ to $(vmin_i^{'}, vmax_i^{'})$ for each of the $i-th \quad (i = 1, 2, \ldots, N)$ object in the image. Here, the values of the two numbers $vmin_i^{'}$ and $vmax_i^{'}$ are randomly determined by satisfying $vmin_i < vmin_i^{'} < vmax_i^{'} < vmax_i$.

(3) Assign a new value $I'_{i,j}$ for the $j-th$ pixel $p_{i,j}$ that belonging to the $i-th$ object $mask_i$ in the RGB image by using the following Eq. 14 and obtain the final synthetic thermal image. In Eq. 14, $I_{i,j}$ denotes the original gray value for pixel $p_{i,j}$. By using Eq. 14, the gray values for the $i-th$ object in the original RGB image are re-assigned within the range of $\left(vmin_i^{'}, vmax_i^{'}\right)$.

$$I'_{i,j} = \left(vmax_i^{'} - vmin_i^{'}\right)/255 \times I_{i,j} + vmin_i^{'}. \quad (14)$$

Fig. 9 illustrates some synthetic thermal images generated from RGB images and some real thermal images taken from

similar scenes. As shown in Fig. 9, these synthetic thermal images can well reflect the thermal infrared characteristics of an object and achieves similar visual results with those real thermal images.

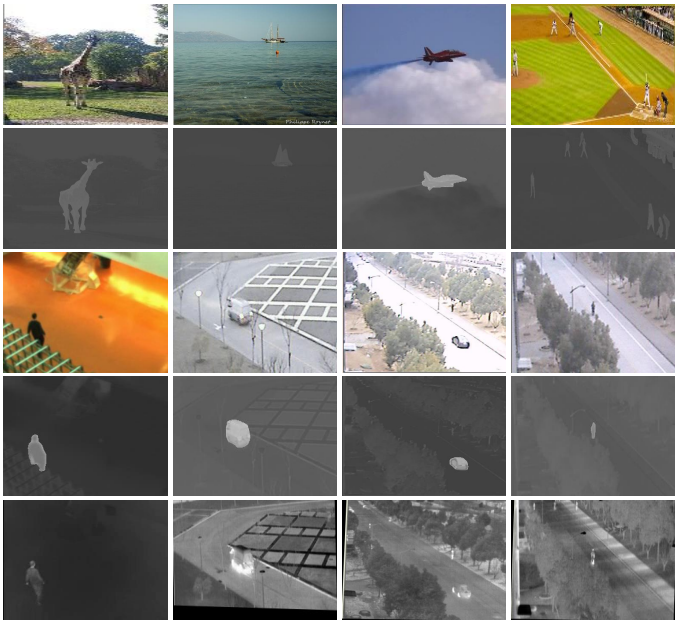

Fig. 9. More examples of our synthetic thermal images generated by our proposed method. The first and third rows are original RGB images, and the second and fourth rows are synthetic thermal images generated by our method. The fifth row is real thermal images corresponding to the third row of RGB images.

B. Generate RGB images from thermal images

Considering the existence of large-scale thermal datasets [52], generating synthetic RGB data from thermal data can also further expand the training data. Moreover, the real thermal data is helpful for the model to learn the effective feature representation of thermal images. Different from generating thermal images from RGB images, which may be seen as a style translation problem, generating RGB images from thermal images is treated as an image colorization task here, since thermal images lack color information. Furthermore, in order to maintain the good color consistency among different frames in a video, we employ the video colorization method in [47] to covert the thermal videos into RGB videos in a video translation way. Examples shown in Fig. 10 demonstrate that the generated RGB images (or videos) have good color consistency and achieve satisfactory visual qualities.

Totally, the newly built LSS dataset contains 12K synthetic thermal images and 3862 synthetic thermal videos generated from COCO dataset [51] and VID dataset [48] by using the proposed semantic-aware image-to-image translation method as well as 969 synthetic RGB videos generated from [52] by using the video colorization method [47].

V. Experiments

In this section, we will first specify the implementation details and experimental setup, then we will compare our proposed tracking algorithm with some states-of-the-arts on three tracking benchmarks, i.e., VOT-RGBT2019 [15], RGBT234 [28] and GTOT [9]. Finally, some ablation studies are made to verify the effectiveness of each proposed component on VOT-RGBT2019 [15]. As in [23], the input size of the template patches and search patches are set to 127 pixels and 255 pixels, respectively. Our tracker is implemented by using Pytorch on PC with Intel-Xeon(R) 4214 CPU (2.2GHz), 64 GB RAM and Nvidia RTX-2080Ti GPUs (11 GB memory).

A. Implementation Details

1) Data augmentation: We apply several data augmentation strategies including blur, scale change and spatial aware sampling strategy in [23]. Following [36], we also apply distractor-aware training to increase the semantic negative samples. In addition, in order to take full advantage of the complementary information between the input RGB and thermal images, we have made some additional image augmentations on the synthetic RGB-T videos. We apply brightness adjustment, contrast adjustment and Gaussian blur, respectively, to randomly reduce the image quality from a certain modality.

2) Anchors setting: For each point, our anchor boxes have 5 aspect ratios, i.e., [0.33, 0.5, 1, 2, 3], and the anchor scale is set to 8. In Siamese-RPN blocks, we determine the correspondences between the anchors and groundtruth boxes based on IoU. Specifically, if the IoU between the anchor and ground-truth box is larger than 0.6, the anchor is determined as a positive one. Meanwhile, if the IoU between the anchor and ground-truth box is less than 0.3, the anchor is determined as a negative one. We collect at most 16 positive samples and 48 negative samples from one image pair.

3) Optimization: Our model is trained in an end-to-end fashion, where the training loss is a weighted combination of multiple loss functions applied in SiamRPN++ [23]. Specifically, we apply the cross-entropy loss as our classification loss and the smooth L_1 loss as our regression loss. We use the SGD optimization algorithm with 0.9 momentum and 0.0005 weight decay to train our model. Notably, the proposed model is trained in two stages. In the first stage, we disable the thermal Siamese network and multi-modal fusion network to construct an unimodal tracking network. We adopt RGB tracking datasets, including ImageNet VID [48], Youtube-BB [54], COCO [51] and ImageNet Det [48], as our basic training datasets. We use a warmup learning rate of 0.001 for the first 5 epochs to train the Siamese Region Proposal networks. For the last 15 epochs, the whole network is end-to-end trained with learning rate exponentially decayed from 0.005 to 0.0005.

In the second stage, we adopt RGB-T tracking datasets and our newly built LSS dataset to train the whole model. The real RGB-T datasets include KAIST [55], RGBT234 [28] and GTOT [9]. In order to keep the real RGB-T data in the training dataset as much as possible and avoid overfitting, we train our network by using LSS, RGBT234 and KAIST when conducting evaluations on GTOT. When testing our tracker on RGBT234, we train the network by using LSS, GTOT and KAIST. We employ LSS, KAIST and 174 videos in RGBT234 as the training dataset when testing our trackers on VOT-RGBT2019 [15]. We fix all the parameters

(a) (b)

Fig. 10. More examples of our generated RGB images. (a) In each row, we present two pairs of RGB-T images spaced 30 frames apart in the same video; (b) The first column is original thermal images, the second column is synthetic RGB images, and the third column is real RGB images corresponding to the first column of thermal images.

TABLE II
TRACKING PERFORMANCE OF DIFFERENT TRACKERS ON VOT BENCHMARK. 'A' AND 'R' DENOTE ACCURACY AND ROBUSTNESS. EAO STANDS FOR EXPECTED AVERAGE OVERLAP. THE NUMBERS WITH RED, GREEN AND BLUE INDICATE THE BEST, SECOND BEST AND THIRD BEST RESULTS, RESPECTIVELY.

Trackers	ATOM	DiMP	SiamRPN++	SiamBAN	GESBTT	CISRDCF	SGT	SiamFT	MPAT	MANet	FSRPN	mfDiMP	SiamDW-T	JMMAC	Ours
A	0.587	0. 601	0.641	0.622	0.616	0.522	0.518	0.630	0.572	0.582	0.636	0.602	0.616	0.665	0.682
R	0.695	0.709	0.648	0.706	0.635	0.690	0.723	0.639	0.724	0.701	0.708	0.804	0.784	0.821	0.757
EAO	0.321	0.327	0.329	0.333	0.290	0.292	0.297	0.310	0.318	0.346	0.355	0.388	0.393	0.482	0.424
FPS	32	40	48	52	-	-	5	43	18	2	33	22	14	4	37

in the RGB Siamese network and train the thermal Siamese network with the learning rate exponentially decayed from 0.0005 to 0.00005. The multi-modal fusion network is trained with the learning rate exponentially decayed from 0.005 to 0.0005. Other parameters are trained with the learning rate exponentially decayed from 0.0005 to 0.00005.

B. Evaluation on Tracking Dataset

In order to evaluate the overall performance, we compare our method with 4 state-of-the-art RGB trackers, including SiamRPN++ [23], SiamBAN [43], ATOM [31] and Dimp [33], as well as 4 RGB-T trackers, including SGT [53], MANet [13], SiamFT [17] and mfDiMP [26], on three challenging datasets, i.e, VOT-RGB2019 [15], GTOT [9] and RGB234 [28]. In addition, we compare our method with nine recent trackers in the official VOT-RGB2019 [15] challenge report. Specifically, for RGB trackers, we only use RGB data to test their performance.

1) VOT-RGBT2019 dataset: VOT-RGBT2019 [15] contains 60 testing sequences. Targets are annotated by rotated rectangles to enable a more thorough localization accuracy. We adopt Accuracy (A), Robustness (R) and Expected Average Overlap (EAO) as in [15] to evaluate different trackers. Accuracy and Robustness reflect the accuracy and robustness of the tracker, while Expected Average Overlap reflects the overall performance of the tracker. Higher values of Accuracy, Robustness and Expected Average Overlap are more desirable for a tracker.

Table II shows the performance of different trackers on VOT-RGBT2019 dataset. SiamCDA improves the second best Siamese tracker, i.e., SiamDW-T [15], by an absolute gain of 3.1% in terms of EAO. In addition, the proposed method achieves the top-ranked performance in terms of Accuracy and the second position in terms of EAO among these trackers. Although in terms of Robustness, the proposed method is still inferior to some RGB-T trackers, such as JMMAC [15], SiamDW-T [15] and mfDiMP [26], but our proposed tracker is much faster, which turns out to be important for real-time applications.

2) GTOT dataset: GTOT [9] contains 50 RGB-T video sequences annotated with seven challenging attributes, including occlusion (OCC), large scale variation (LSV), fast motion (FM), low illumination (LI), thermal crossover (TC), small object (SO) and deformation (DEF). We adopt the precision rate (PR) and success rate (SR) as in [13] for quantitative performance evaluation. PR is the percentage of frames whose output locations are within a threshold distance to the groundtruth. We set the threshold to be 5 here because the target objects are generally small in GTOT. SR is the percentage of the frames whose overlap ratios between the output bounding boxes and their groundtruth bounding boxes are larger than a threshold, and we compute the SR score by the area under curves.

The attribute-based comparisons also show the capability of our proposed tracker in handling those challenging situations. As shown in Table III, our tracker obtains the best performance

TABLE III
ATTRIBUTE-BASED PRECISION RATE AND SUCCESS RATE (PR/SR) OBTAINED BY USING DIFFERENT TRACKERS ON GTOT DATASET. THE NUMBERS WITH RED, GREEN AND BLUE COLOR INDICATE THE BEST, SECOND BEST AND THIRD BEST RESULTS, RESPECTIVELY.

Method	SiamBAN [43]	SiamRPN++ [23]	ATOM [31]	DiMP [33]	SiamFT [17]	SGT [53]	mfDiMP [26]	MANet [13]	Ours
OCC	67.2/54.9	70.3/58.7	67.4/55.1	75.7/63.8	75.3/58.6	81.0/56.7	80.7/64.3	88.2/69.6	82.2/69.4
LSV	78.3/64.2	76.5/64.3	78.9/64.2	81.4/69.0	79.7/61.4	84.2/54.7	90.5/73.9	86.9/70.6	91.5/74.8
FM	74.3/62.0	75.9/65.9	74.8/63.0	78.9/68.0	72.1/60.1	79.9/55.9	81.3/68.7	87.9/69.4	86.6/72.0
LI	66.8/56.0	68.9/58.3	68.3/58.4	69.8/61.1	78.6/63.6	88.4/65.1	83.0/70.4	91.4/73.6	92.4/76.4
TC	76.3/61.0	76.6/64.0	79.0/63.3	84.2/68.7	76.0/59.3	84.8/61.5	80.4/65.2	88.9/70.2	82.6/68.5
DEF	66.1/55.5	71.0/59.3	69.1/58.8	69.9/59.9	72.5/61.9	91.9/73.3	80.7/67.1	92.3/75.2	87.9/72.7
SO	79.3/59.3	82.2/64.7	83.7/62.9	84.2/64.0	79.3/59.3	91.7/61.8	87.4/69.1	93.2/70.0	88.4/71.3
ALL	71.7/59.3	72.5/61.7	72.6/61.2	75.7/64.9	75.8/62.3	85.1/62.8	83.6/69.7	89.4/72.4	87.7/73.2
FPS	52	48	32	40	43	5	22	2	37

TABLE IV
ATTRIBUTE-BASED PRECISION RATE AND SUCCESS RATE (PR/SR) OBTAINED BY USING DIFFERENT TRACKERS ON RGBT234 DATASET. THE NUMBERS WITH RED, GREEN AND BLUE COLOR INDICATE THE BEST, SECOND BEST AND THIRD BEST RESULTS, RESPECTIVELY.

Method	SiamBAN [43]	SiamRPN++ [23]	ATOM [31]	DiMP [33]	SiamFT [17]	SGT [53]	MANet [13]	mfDiMP [26]	Ours
NO	82.3/61.3	83.8/64.2	81.3/61.7	83.2/63.8	84.8/62.0	87.7/55.5	88.7/64.6	88.5/65.0	88.4/66.4
PO	72.7/53.2	73.5/54.3	77.9/55.9	80.4/58.5	72.7/50.6	77.9/51.3	81.6/56.6	83.7/59.9	84.2/63.9
HO	57.0/39.5	59.4/43.4	63.5/43.3	64.2/45.0	57.6/40.7	59.2/39.4	68.9/46.5	69.6/46.6	66.2/48.7
LI	59.0/41.3	59.3/42.4	63.4/44.7	62.9/44.5	68.8/47.4	70.5/46.2	76.9/51.3	78.0/54.1	81.8/61.2
LR	63.5/44.2	66.4/46.5	66.1/44.0	66.0/44.8	69.6/46.5	75.1/47.6	75.7/51.5	75.9/49.2	70.9/49.9
TC	71.0/50.1	70.6/53.0	78.3/57.2	81.7/60.4	70.7/50.3	76.0/47.0	75.4/54.3	75.7/52.7	67.4/47.7
DEF	68.5/51.1	69.5/53.2	70.4/52.1	73.3/54.5	69.7/51.8	73.7/53.5	72.0/52.4	77.0/56.0	77.9/59.2
FM	56.8/39.1	65.3/46.9	70.3/48.0	69.0/48.2	62.0/42.2	67.7/40.2	69.4/44.9	73.5/50.2	61.4/45.3
SV	72.3/53.5	72.7/55.5	78.8/57.2	77.8/58.2	71.1/50.5	69.2/43.4	77.7/54.2	81.7/58.7	77.7/59.3
MB	61.2/44.2	64.5/48.7	71.1/51.6	72.2/53.0	59.0/43.3	64.7/43.6	72.6/51.6	74.2/52.4	63.6/47.9
CM	64.3/47.0	66.4/49.9	66.1/48.7	70.1/52.2	63.2/45.7	66.7/45.2	71.9/50.8	76.4/54.2	73.3/54.7
BC	57.8/38.0	57.8/39.3	58.0/38.3	59.2/39.4	60.5/40.3	65.8/41.8	73.9/48.6	71.8/45.4	74.0/52.9
ALL	68.1/49.1	69.7/51.7	72.7/51.8	74.3/54.0	68.8/48.6	72.0/47.2	77.7/53.9	78.9/55.4	76.0/56.9

with 73.2% in success score and the second best performance with 87.7% in precision score. Compared with the second best Siamese tracker, i.e., SiamFT, our algorithm achieves 10.9% improvement in success and 11.9% improvements in precision. Compared with the most recent tracker, i.e., MANet, our tracker achieves the tracking performance of 1.7% lower in PR but 0.8% higher in SR. This demonstrates that we have reduced and even eliminated the gap in tracking performance between the RGB-T Siamese trackers and some other state-of-the art RGB-T trackers. Moreover, our proposed tracker achieves much efficient running speed. As shown in Table III, our tracker achieves the leading performance at a real-time running speed of 37 FPS.

3) RGBT234 dataset: RGBT234 [28] is a large-scale RGB-T tracking dataset. It contains 234 pairs of visible and thermal videos. RGBT234 contains 12 annotated attributes, including no occlusion (NO), partial occlusion (PO), heavy occlusion (HO), low illumination (LI), low resolution (LR), thermal crossover (TC), deformation (DEF), fast motion (FM), scale variation (SV), motion blur (MB), camera moving (CM) and background clutter (BC). Here, we still adopt the precision rate (PR) and success rate (SR) to quantitatively evaluate the performance.

Tracking results are shown in Table IV, which indicate that our tracker achieves 76.0%/56.9% in PR/SR, and has 6.3%/5.2% promotion over the second best Siamese tracker, i.e., SiamRPN++. To further demonstrate the effectiveness of our tracker, we provide the attribute-based performance on RGBT234 in Table IV. From Table IV, we can observe that the proposed method significantly outperforms other trackers in most cases. First, most of these trackers perform very well in cases of no occlusions, but drop a lot when partial or heavy occlusions happen. Our tracker still keeps high tracking performance in these cases. This may owe to the full use of the complementary information between RGB and thermal images in our proposed tracker. Second, in cases of low illumination and low resolution, our tracker outperforms most trackers, especially significantly outperforms those RGB trackers. This further demonstrates the effectiveness of our proposed model on using multi-modal information to some extent. Third, in the cases of thermal crossover, our tracker does not perform well enough, which indicates that some differences may still exist between the synthetic RGB-T videos we have generated and real RGB-T videos. Finally, in the cases of fast motion and motion blur, those Siamese trackers, including our proposed tracker, do not achieve satisfactory tracking results, because these trackers entirely rely on offline training and local search.

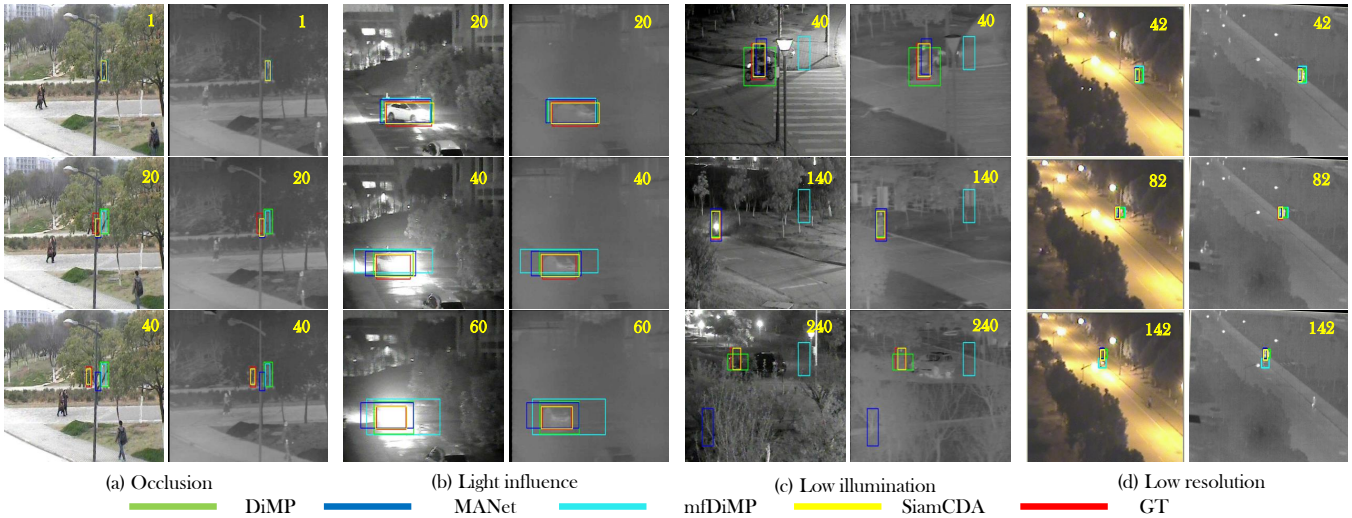

Fig. 11. Visual comparisons of our proposed tracker with another three state-of-the-art trackers on four video sequences, i.e., baketballwaliking, carLight, elelckbike10 and WalingNig.

4) Qualitative performances: The visual comparisons between our proposed method and other state-of-the-art trackers, including DiMP [33], MANet [13] and mfDiMP [26], are shown in Fig. 11. Our approach performs obviously better than other methods in various challenging scenarios, including heavy occlusion, light influence, low illumination and low resolution. For instance, in Fig. 11 (a), our method performs well in presence of heavy occlusions, while other trackers lose the target when occlusion happens. In Fig. 11 (b) and Fig. 11 (c), targets are invisible in RGB image but visible in the thermal images. Compared to those RGB trackers, our approach performs obviously better. This demonstrates that our tracker may make full use of multi-modal information. In Fig. 11 (d), which has low resolution attributes, our tracker also performs better than other methods.

C. Ablation Study

TABLE V
TRACKING RESULTS OBTAINED BY USING DIFFERENT MODULES IN OUR TRACKER.

Fine-tune	FPN	CA-MF	DAS	EAO	A	R	△EAO
				0.346	0.670	0.651	0
√				0.364	0.668	0.685	+0.018
√	√			0.365	**0.690**	0.709	+0.001
√	√	√		0.375	0.682	0.724	+0.010
√	√	√	√	**0.424**	0.682	**0.757**	+0.049

1) Model Architecture: To validate the effectiveness of different components (or modules) in our proposed model, we first construct a simplified version of our proposed tracker as the baseline by retaining the feature extractor and region proposal networks and by replacing the CA-MF module with some simple convolutional layers. The thermal Siamese network and RGB Siamese network also share the same weights in the simplified version. Then, different modules or strategies are added into the baseline. Table V provides the tracking performance of our tracker by using different modules on VOT-RGBT2019 [15]. As shown in Table V, the baseline tracker can achieve an EAO of 0.346. When we fine-tune the thermal Siamese network, the EAO score is increased to 0.364. By further adding FPNs to the baseline tracker, the EAO can be improved to 0.365. When we subsequently use the CA-MF modules, the EAO is increased to 0.375 and the robustness is increased from 0.709 to 0.724. In the end, we achieve an EAO score of 0.424 by using the proposed DAS module, which surpasses the baseline by a large margin of 7.8%. This indicates that jointly exploring these modules makes our method not only robust but also accurate.

2) Multi-modal fusion: To further validate the effectiveness of our proposed CA-MF module, we employ 4 fusion strategies in our tracker to fuse multi-modal features, construct 4 different versions of our approach for comparative analysis, including: 1) Element-wise summation. 2) Concatenation. 3) Contented-based fusion strategy. 4) The proposed CA-MF module. The only difference between the last two methods is that the contented-based fusion strategy directly fused the weighted unimodal RGB and thermal features via concatenation. According to the experimental data in Table VI, it can be seen that the proposed CA-MF module outperforms the other fusion modules significantly.

TABLE VI
TRACKING RESULTS OBTAINED BY USING DIFFERENT FUSION MODULES.

Fusion module	EAO	A	R	△EAO
Element-wise summation	0.375	0.682	0.724	0
Concatenation	0.379	**0.686**	0.720	+0.004
Content-based fusion strategy	0.383	0.680	0.727	+0.004
CA-MF module	**0.424**	0.682	**0.757**	+0.041

3) Training Data: We also verify the validity of training data, including the proposed two-stage training strategy and synthetic data augmentation strategy in this subsection. Firstly, we only use real RGB-T videos to train our model. Our tracker can achieve an EAO of 0.329. Secondly, the proposed two-stage training strategy described in Subsection V-A3 can

TABLE VII
IMPACTS OF DIFFERENT TRAINING STRATEGIES ON VOT2019RGBT.

Real RGB-T data	Pretrain	Synthetic RGB-T data	Data augmentation	EAO	A	R	△EAO
√				0.329	0.661	0.623	0
√	√			0.336	0.655	0.671	+0.007
√	√	√		0.396	**0.686**	0.720	+0.060
√	√	√	√	**0.424**	0.682	**0.757**	+0.028

improve the tracking performance to some extent. As shown in Table VII, the EAO score is increased to 0.336. Thirdly, by enlarging the training data with our synthetic data, the performance is increased by a margin of 6.0%. Finally, performing some additional data enhancements on the synthetic videos makes the synthetic data closer to the real data and thus further improves tracking performance.

VI. CONCLUSION

In this paper, we present a new RGB-T Siamese tracker. Owing to the collaboration of some newly designed modules, our proposed tracker achieves state-of-art performance with real-time running speed. Especially, by virtue of the proposed CA-MF, our tracker can make full use of the complementary advantages of multi-modal features, thus able to achieve satisfactory results in some challenging conditions, such as heavy occlusion and illumination variations. Thanks to the proposed DAS, our tracker shows good robustness against some distractors, i.e., semantic backgrounds. Finally, the feature representation ability of our tracker is significantly enhanced by jointly employing the newly built synthetic RGB-T tracking dataset and some real RGB-T tracking datasets during the training phase. This further improves the tracking performance of our proposed tracker. Extensive experiments on three benchmark datasets demonstrate that our proposed tracker significantly outperforms those existing RGB-T Siamese trackers. Compared with some other state-of-the-arts, our proposed tracker performs competitively and even slightly better in tracking accuracy but shows obvious superiorities in tracking speed.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 61773301.

REFERENCES

[1] K. Lee and J. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1429–1438, 2015.

[2] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image and Vision Computing*, vol. 22, no. 2, pp. 143–155, 2004.

[3] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, 2000.

[4] C. Li, C. Zhu, J. Zhang, B. Luo, X. Wu, and J. Tang, "Learning local-global multi-graph descriptors for RGB-T object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 2913–2926, 2018.

[5] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with RGB-D data," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 2101–2107.

[6] Q. Zhang, N. Huang, L. Yao, D. Zhang, C. Shan, and J. Han, "Rgb-t salient object detection via fusing multi-level cnn features," *IEEE Transactions on Image Processing*, vol. 29, pp. 3321–3335, 2020.

[7] Q. Zhang, T. Xiao, N. Huang, D. Zhang, and J. Han, "Revisiting feature fusion for rgb-t salient object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[8] C. Ó. Conaire, N. E. O'Connor, and A. Smeaton, "Thermo-visual feature fusion for object tracking using multiple spatiogram trackers," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 483–494, 2008.

[9] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin, "Learning collaborative sparse representation for grayscale-thermal tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5743–5756, 2016.

[10] C. Li, X. Sun, X. Wang, L. Zhang, and J. Tang, "Grayscale-thermal object tracking via multitask laplacian sparse representation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 673–681, 2017.

[11] C. Li, X. Wu, N. Zhao, X. Cao, and J. Tang, "Fusing two-stream convolutional neural networks for RGB-T object tracking," *Neurocomputing*, vol. 281, pp. 78–85, 2018.

[12] Y. Zhu, C. Li, B. Luo, J. Tang, and X. Wang, "Dense feature aggregation and pruning for RGBT tracking," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 465–472.

[13] C. Li, A. Lu, A. Zheng, Z. Tu, and J. Tang, "Multi-adapter RGBT tracking," in *Proceedings of the IEEE Conference on Computer Vision Workshops*, 2019.

[14] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[15] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Cehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg, A. Eldesokey, J. Kapyla, G. Fernandez, A. Gonzalez-Garcia, A. Memarmoghadam, A. Lu, A. He, A. Varfolomieiev, A. Chan, A. Shekhar Tripathi, A. Smeulders, B. Suraj Pedasingu, B. Xin Chen, B. Zhang, B. Wu, B. Li, B. He, B. Yan, B. Bai, B. Li, B. Li, B. Hak Kim, and B. Hak Ki, "The seventh visual object tracking vot2019 challenge results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[16] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proceedings of the IEEE Conference on Computer Vision Workshops*, 2016, pp. 850–865.

[17] X. Zhang, P. Ye, S. Peng, J. Liu, K. Gong, and G. Xiao, "SiamFT: An RGB-Infrared fusion tracking method via fully convolutional Siamese networks," *IEEE Access*, vol. 7, p. 122122–122133, 2019.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

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

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[21] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[24] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[25] X. Zhang, P. Ye, S. Peng, J. Liu, and G. Xiao, "Dsiammft: An RGB-T fusion tracking method via dynamic Siamese networks using multi-layer feature fusion," *Signal Processing: Image Communication*, vol. 84, p. 115756, 2020.

[26] 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.

[27] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

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

[29] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 472–488.

[30] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2017, pp. 6638–6646.

[31] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2019, pp. 4660–4669.

[32] R. Yao, G. Lin, C. Shen, Y. Zhang, and Q. Shi, "Semantics-aware visual object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1687–1700, 2018.

[33] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6182–6191.

[34] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: Visual tracking by re-detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6578–6588.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2015, pp. 91–99.

[36] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 101–117.

[37] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[38] D. Li, F. Porikli, G. Wen, and Y. Kuai, "When correlation filters meet Siamese networks for real-time complementary tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 509–519, 2019.

[39] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4591–4600.

[40] C. Zhu, Y. He, , and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 840–849.

[41] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.

[42] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines." in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 549–12 556.

[43] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, 2020, pp. 6668–6677.

[44] J. F. B. L. W. H. Zhipeng Zhang, Houwen Peng, "Ocean: Object-aware anchor-free tracking," in *Proceedings of the European Conference on Computer Vision*, 2020.

[45] B. Yan, D. Wang, H. Lu, and X. Yang, "Cooling-shrinking attack: Blinding the tracker with imperceptible noises," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[46] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic Siamese network for visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1763–1771.

[47] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3753–3761.

[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[50] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[51] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[52] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, W. Liu, and Y. Liang, "Multi-task driven feature models for thermal infrared tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 604–11 611.

[53] C. Li, N. Zhao, Y. Lu, C. Zhu, and J. Tang, "Weighted sparse representation regularized graph learning for RGB-T object tracking," in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017, pp. 1856–1864.

[54] E. Real, J. Shlens, S. Mazzocchi, P. Xin, and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5296–5305.

[55] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1037–1045.

Tianlu Zhang received the B. S. degree from Xi'an Shiyou University, Xi'an, China, in 2018. He is currently pursuing the Ph.D. degree in School of Mechano-Electronic Engineering, Xidian University, China. His research interests include deep learning and multimodal image processing in computer vision.

Xueru Liu received his B. S. degree from Xidian University, Xi'an, China, in 2019. She is currently pursuing the M.S. degree in School of Mechano-Electronic Engineering, Xidian University, China. Her current research interests include multimodal image processing and deep learning.

Qiang Zhang received the B.S. degree in automatic control, the M.S. degree in pattern recognition and intelligent systems, and the Ph.D. degree in circuit and system from Xidian University, China, in 2001,2004, and 2008, respectively. He was a Visiting Scholar with the Center for Intelligent Machines, McGill University, Canada. He is currently a professor with the Automatic Control Department, Xidian University, China. His current research interests include image processing, pattern recognition.

Jungong Han is currently a Full Professor and Chair in Computer Science at Aberystwyth University, UK. His research interests span the fields of video analysis, computer vision and applied machine learning. He has published over 180 papers, including 40+ IEEE Trans and 40+ A* conference papers.