IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 20, 2019, accepted November 22, 2019, date of publication December 9, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958405

Deep Motion-Appearance Convolutions for Robust Visual Tracking

HAOJIE LI[1], SIHANG WU[1], SHUANGPING HUANG[1], KIN-MAN LAM[2], (Member, IEEE), AND XIAOFEN XING[1]

[1]College of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China
[2]College of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong

Corresponding author: Shuangping Huang (huangshuangping@gmail.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61673182, Grant 61936003, and Grant 61702192, in part by the Guangdong-Natural Science Foundation under Grant 2017A030312006, and in part by the Science and Technology Program of Guangzhou, China under Grant 201902010069, Grant 201707010160, and Grant 201704020134.

ABSTRACT Visual tracking is a challenging task due to unconstrained appearance variations and dynamic surrounding backgrounds, which basically arise from the complex motion of the target object. Therefore, the information and the correlation between the target motion and its resulting appearance should be considered comprehensively to achieve robust tracking performance. In this paper, we propose a deep neural network for visual tracking, namely the Motion-Appearance Dual (MADual) network, which employs a dual-branch architecture, by using deep two-dimensional (2D) and deep three-dimensional (3D) convolutions to integrate the local and global information of the target object's motion and appearance synchronously. For each frame of a tracking video, 2D convolutional kernels of the deep 2D branch slide over the frame to extract its global spatial-appearance features. Meanwhile, 3D convolutional kernels of the deep 3D branch are used to collaboratively extract the appearance and the associated motion features of the visual target from successive frames. By sliding the 3D convolutional kernels along a video sequence, the model is able to learn the temporal features from previous frames, and therefore, generate the local patch-based motion patterns of the target. Sliding the 2D kernels on a frame and the 3D kernels on a frame cube synchronously enables a better hierarchical motion-appearance integration, and boosts the performance for the visual tracking task. To further improve the tracking precision, an extra ridge-regression model is trained for the tracking process, based not only on the bounding box given in the first frame, but also on its synchro-frame-cube using our proposed Inverse Temporal Training method (ITT). Extensive experiments on popular benchmark datasets, OTB2013, OTB50, OTB2015, UAV123, TC128, VOT2015 and VOT2016, demonstrate that the proposed MADual tracker performs favorably against many state-of-the-art methods.

INDEX TERMS Visual tracking, 3D convolutional kernels, motion-appearance.

I. INTRODUCTION

Visual tracking is a fundamental task in the field of computer vision, where a target, specified by a bounding box in the first frame, is to be tracked in the subsequent frames. Although numerous algorithms have been proposed for visual tracking, it remains a highly challenging problem, especially when the target object in the video suffers from drastic deformation, rotation, scale and illumination variations, etc.

To address the difficulties brought by target-appearance changes, researchers have proposed convolutional neural network (CNN)-based approaches [1]–[4], in the hope of increasing the robustness of appearance representation through extracting deep features, instead of low-level hand-crafted features. These methods either combine with traditional frameworks, such as Discriminative Correlation Filters (DCFs) [4]; achieve elegant architectural design, such as that based on the Siamese network algorithm series [5]; or integrate with a Region Proposal Network (RPN) technique for object detection [3]. Regardless of the promising performance brought by the discriminative power of deep features, the performance of tracking-by-detection methods is limited, because they fail to incorporate motion and inter-frame information from videos [6]. As a result, these methods have their

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Mehmood.

performance degraded, when the visual target shows in-plane and out-of-plane rotations, dramatic scale variations, severe occlusion, etc.

To take motion features into consideration for visual tracking, representative methods, including optical flow and Recurrent Neural Network (RNN), are adopted alone, or in combination with a deep tracker [7], [8]. However, these methods collect motion information from only consecutive pairs of frames, requiring per-pixel correspondence estimation and localization, which result in dense flow field and high computational complexity. In addition, traditional optical flow methods have limited performance, due to the brightness constancy constraint, which is rarely the case in complex video scenes. Moreover, optical flow can only capture low-level motion patterns, which are not robust to the diverse changes of the targets' motion in videos. More recently, some researchers proposed the deep optical flow network [8]–[10], which integrates optical motion information learned by deep networks to improve the computation efficiency and enhance the semantic representation of motion features. However, the tracking performance is not prominent [9]. RNN-based methods consider temporal dependencies and model the temporal state transitions over the frames in a video. This class of method usually has an RNN placed on the top of a deep convolutional network, in which the deep feature of a frame is taken as the input of an RNN time cell. Nevertheless, this structure fails to synchronously model the inter-relationship of the spatial-temporal characteristics in videos.

In this paper, we propose the Motion-Appearance Dual network (MADual), a new dual-branch architecture based on deep 2D and 3D convolutions to improve the feature representation and tracking accuracy. The proposed architecture achieves the spatial-temporal information fusion on both the local and global scales. First, the local fusion of spatial and temporal information is carried out by the 3D convolutional operations. Unlike 2D convolutions, which only capture the spatial features in a single frame, 3D convolutions also span the temporal dimension - i.e. they capture appearance features across multiple consecutive frames. In this manner, the 3D convolutional operations enable the integration of temporal information into the learned deep features. Meanwhile, deep 2D convolutional operations are used to capture the appearance features of the whole target in the current frame. After both the 2D and 3D branches have captured the corresponding features separately, the spatial-temporal fusion on a global scale is performed by concatenating the features from the two branches. To further improve the tracking precision, we also propose an Inverse Temporal Training (ITT) mechanism, which enhances the regression model of our tracker. Finally, we conduct extensive experiments to evaluate our method under various challenging scenarios.

The main contributions of this paper are as follows:

1) We develop a new dual-branch network based on deep 2D and 3D convolutions for visual tracking, which enables a local-to-global integration of the target's motion and appearance information. Thus, it can handle the challenges, such as dramatic scale variations, significant appearance deformation and rotation, etc.

2) A deep 3D network branch, which is more robust to motion noise, is used to learn the semantic-level motion features. To the best of our knowledge, this is the first work to introduce a deep 3D convolutional network for object tracking.

3) We propose an Inverse Temporal Training (ITT) strategy, for introducing deep motion features to the online tracking process, which can assist in training the regressor to obtain a more accurate target location.

4) Extensive experiments were conducted on seven benchmark datasets: OTB2013 [11], OTB2015 [12], OTB50 [12], UAV123 [13], TC-128 [14], VOT2015 [15], and VOT2016 [6], which demonstrate that the proposed MADual tracker performs favorably against existing state-of-the-art methods. The source code and model are obtained from the Github repository.[1]

II. RELATED WORK

We will give a brief review of object tracking, based on three approaches related closely to our proposed work: tracking by deep neural networks (DNNs), tracking by detection, and tracking based on spatial-temporal information.

A. TRACKERS BASED ON DEEP NEURAL NETWORKS

Recently, due to their superior representation power, deep features have been widely employed to boost the performance in visual tracking. Since DCFs provides an excellent framework for tracking [4], [16], a common and important trend is the combination of the DCF framework and CNN features [4], [17]–[20]. For example, DeepSRDCF [4] used deep features in SRDCF [21], and achieved good performance. In [18], the features from different layers of a pretrained CNN, such as VGG [22], are concatenated, and then fed into a correlation filter. C-COT [19] and ECO [20] are trackers proposed based on continuous convolutional filters. Based on ECO [20], CFWCR [23] normalizes the respective features extracted from different layers to obtain a more robust result.

Another group of deep trackers are characterized by using the Siamese network architecture [5], [10], [24]–[28], in which a similarity comparison strategy is required to perform a template match. By comparing the ground-truth patch of a target object with the candidate patches within a search window in the current frame, the most similar patch is considered the target. In particular, Bertinetto et al. [5] proposed a tracker based on the region-wise feature similarity between two successive frames. In [29], a fast tracker was proposed by learning an agent to decide whether an object is located with high confidence in an early layer. In [30], the exemplars used for object tracking were adjusted by means of online updating. In [5], a two-branch Siamese network was proposed, with one branch for semantic learning and the other for appearance

[1] https://github.com/DLCV-HUANG/MADual

learning. In addition, RASNet [27] was introduced with three different attention mechanisms to enhance the tracking performance, and SiamRPN [3] integrated the region-proposal network as the backend to improve the efficiency of tracking objects with different scales. The GOTURN [24] tracker learns a deep regression network to compare crops from a search region in the current frame to the target in the previous frame. SiamVGG employed more advanced networks to achieve better discrimination capability and more accurate object tracking.

From the above trackers, deep CNNs can learn effective high-level semantic representations of object appearance for locating desired targets, and enhance the performance of existing trackers, irrespective of whether correlation filters or Siamese networks are used. These methods are constantly evolving and overcoming the specific limitations of CNN methods for object tracking. One example is to use offline training to alleviate high computational complexity and achieve reliable real-time performance [21], [31]. Another example is to employ more advanced networks, such as AlexNet, VGGNet, and ResNet, for better discrimination capability, and eventually improve the tracking accuracy [1], [4], [5], [28].

Despite the success of DNN-based trackers, they can only achieve sub-optimal tracking performance, as CNNs are only used to enhance the appearance features and hardly benefit from the motion and inter-frame information. In fact, visual trackers can be easily disturbed by similar objects or unseen object's appearances if only the appearance cue is used, resulting in relatively uncertain location predictions in the current frame. In this paper, we additionally introduce a 3D deep network to capture semantic-level motion features, and propose a way to blend both the appearance and motion information to achieve effective target detection and tracking in the current frame, even under challenging conditions. By using a deep network of multiple 3D layers, the semantic-level motion features, which are more robust to varying target appearance due to motion than directly using the low-level motion features obtained from various motion models [9], can be learned.

B. DETECTION-BASED TRACKERS

The tracking-by-detection approaches have been commonly used by most of the popular object trackers in recent years, in which tracking is formulated as the detection problem in each frame [1], [32]–[39]. These approaches emphasize object appearance modeling to decide whether an image patch is a target object or not. Among these approaches, the generative method describes the target appearance using a generative model and searches for the target region that best fits the model. Example algorithms include sparse representation [33], density estimation [34], and incremental subspace learning [35]. Discriminating methods aim to build a classifier that distinguishes the target object from the background. These tracking algorithms typically learn the decision boundary based on multiple instance learning [36], P-N learning [32], online boosting [37], structured output SVMs [40], domain adaptation [1], random forests [38], and ensemble learning [39].

Among the numerous tracking-by-detection algorithms, [1] is one of the most popular CNN-based trackers with state-of-the-art accuracy, which proposed the multi-domain convolutional neural network. Therein, the shared layers of the architecture are pretrained using a large set of videos with tracking ground-truths to obtain a generic target representation. The multiple branches of the domain-specific layers are responsible for binary classification to identify the target in each domain. Following the work of [1], Jung et al. [41] presented a real-time version, striving to differentiate foreground instances across multiple domains and learn a more discriminative embedding of target objects with similar semantics by the introduction of a new loss term. To summarize, the existing detection-based trackers can achieve results only as accurately as those models based on the appearance variations of target objects. In this paper, we go beyond the traditional detection model, with the motion features synergistically integrated with the appearance features to locate the target in the current frame. Our argument is that target objects in previous multiple consecutive frames can be linked together based on temporal information, which carries information about the underlying motion pattern of the targets to be tracked. The motion pattern characterizes the temporal smoothness. When the appearance variations are severe, such that the appearance model cannot be fitted; or when a distractor with similar appearance to the target object occurs, using only the appearance model will probably result in detection and tracking errors. In this paper, motion feature is introduced to enrich the representation of the target object in a current frame, so as to achieve a more accurate estimation of the target position.

C. SPATIAL-TEMPORAL TRACKER

The lack of temporal information greatly degrades the tracking performance when challenges, such as partial occlusion and deformation, etc., occur. To address this problem, RNNs [30], [42]–[44] were introduced to extract time-contextual information to boost tracking performance. In RATM [30], an RNN with an attentional mechanism is used to predict the position of a target at specific time instances, given a real-valued hidden state vector. The state vector summarizes the predictions of previous time steps. The RFL [42] algorithm captures both the spatial and temporal information of a sequence for visual tracking using convolutional LSTM. Except for RNN, the optical-flow methods have always been adopted to obtain the flow information from two consecutive image frames, capturing the motion patterns [8], [9]. For example, DMSRDCF [9] fused handcrafted and deep-appearance features with motion features, which were learned based on the optical-flow network in a DCF-based framework [21]. FlowTrack [8] formulated the optical flow estimation in an end-to-end tracking framework, and modeled the motion during training. Although the performance of all

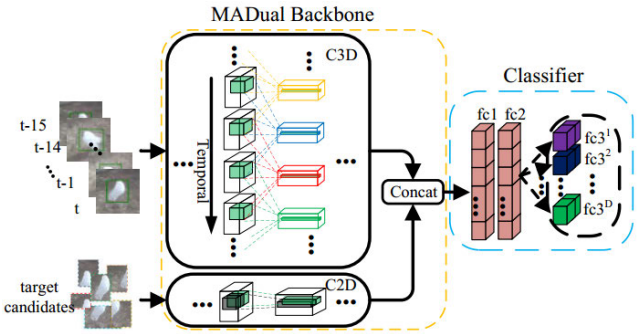

FIGURE 1. Offline architecture.

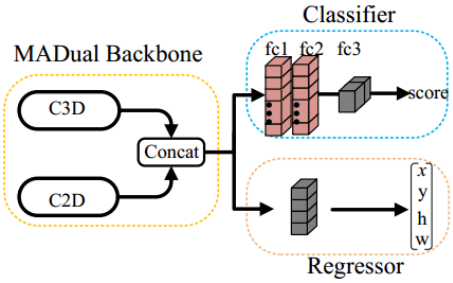

FIGURE 2. Regression-based tracking architecture.

the above-mentioned trackers benefits from the introduction of the temporal or motion information, they have obvious shortcomings. An RNN is focused on temporal dependencies across frames instead of modeling spatial-temporal information synergistically. The optical-flow method directly models motion pattern based on per-pixel correspondence and motion estimation at a low-level manner, thus it is computationally intensive and lacks deep semantic-level motion features. In contrast, our proposed convolutional 3D (C3D) network is well-suited for joint spatial-temporal feature learning, which motivates us to propose a C3D-based tracking framework.

Actually, 3D convolutional models have demonstrated their outstanding representation power in a wide range of video analysis tasks [45]–[50]. In [45], 3D convolutions were first proposed to recognize human action, aimed at capturing the motion information encoded in multiple adjacent frames. In [51], the C3D methods outperformed state-of-the-art video classifiers, in which a homogeneous architecture with small 3×3×3 convolution kernels in all layers was designed to obtain the best performing classifier. [46] applied the 3D convolutional network to capture the local and global temporal structure of short videos to produce descriptions. CDC [48] and S-CNN [49] addressed temporal action localization in untrimmed long videos, utilizing the features from deep 3-dimensional convolutional networks (3D ConvNets). Despite the success of C3D in video analysis tasks, such as action recognition, temporal localization, classification, etc., no research work can be found to explore C3D in visual tracking, to the best of our knowledge. To take the advantage of the motion feature from the C3D network, in this paper we propose the ITT strategy to improve the regression process for obtaining more precise target positions.

III. METHOD

In this section, we describe the proposed dual-branch network-based MADual tracker, which consists of an offline architecture (Figure 1), an online tracking architecture based on ridge regression (Figure 2), and the achieved hierarchical local-global motion-appearance integration through the network design.

A. OFFLINE ARCHITECTURE

As illustrated in Figure 1, our offline architecture consists of two branches, forming its backbone network. For the two branches, a deep convolutional 2D (C2D) network is used to extract the appearance representation of the target object from the current frame, .i.e. frame t, and C3D is for learning the semantic-level motion features between successive frames. Subsequently, the outputs of these two branches are concatenated, followed by two fully connected layers, i.e., $fc1$ and $fc2$, each with 2048 output units. Finally, the network is equipped with D branches, i.e. $fc3^1 - fc3^D$, with each branch corresponding to a different domain, serving as the last fully connected layer. Each of the D branches contains a binary classification layer with the SoftMax cross entropy loss, which is responsible for distinguishing the target from the background in each domain [1]. Note that we refer to $fc3^1 - fc3^D$ as domain-specific layers and all the preceding layers as shared layers [1].

The architecture of our C2D branch is shown on the left of Table 1. It receives a 107×107 RGB input and contains three convolutional layers (Conv1-3). The configuration of these convolution layers is identical to the corresponding parts of VGG-M [52], which is pretrained on the ImageNet [53] dataset. In our design, VGG-M is tailored to be a substantially simplified version to serve as the deep 2D convolutional branch. This is because when the network becomes deeper, class semantics will be overemphasized and the lower-level features tend to be diluted. This is not suitable for handling domain adaptation in visual tracking.

TABLE 1. C3D deep structure.

Type	C2D	Type	C3D
Conv1	$[96 \times 7 \times 7]$	Conv1	$[64\times3\times(3\times3)]$
		Maxpool	$1\times(2\times2)$
Maxpool	3×3	Conv2	$[128 \times 3 \times (3 \times 3)]$
		Maxpool	$2 \times (2 \times 2)$
Conv2	$[256 \times 5 \times 5]$	Conv3_x	$\begin{bmatrix} 256 \times 3 \times (3 \times 3) \\ 256 \times 3 \times (3 \times 3) \end{bmatrix}$
		Maxpool	$2 \times (2 \times 2)$
Maxpool	3×3	Conv4_x	$\begin{bmatrix} 512 \times 3 \times (3 \times 3) \\ 512 \times 3 \times (3 \times 3) \end{bmatrix}$
		Maxpool	$2 \times (2 \times 2)$
Conv3	$[512 \times 3 \times 3]$	Conv5_x	$\begin{bmatrix} 512 \times 3 \times (3 \times 3) \\ 512 \times 3 \times (3 \times 3) \end{bmatrix}$
		Maxpool	$2 \times (1 \times 1)$

The architecture of our C3D is illustrated in Figure 3, and the detailed network parameters are listed on the right of Table 1. From Figure 3, the network has totally 8 convolutional layers and 5 pooling layers, and is tailored based on the original network structure proposed in [51], by removing the last two fully connected layers. We employ this structure for two reasons. The first one is that this structure has been empirically proven to be a good architecture for the video analysis tasks [54], [55]. The second reason is that the model pretrained on a large video dataset, i.e. the Sports-1M [56], is available publicly. The 8 convolutional layers can be categorized into 5 types, with different numbers of filters for the different types. These 5 types are denoted as Conv1, Conv2, Conv3_x, Conv4_x, and Conv5_x, respectively, as tabulated in Table 1, and the corresponding number of filters are 64, 128, 256, 512, and 512. All the 3D convolutional kernels are homogeneously set as $3 \times 3 \times 3$, with stride $1 \times 1 \times 1$, which is the best setting proved by [54], [55]. All the 3D pooling layers are $2 \times 2 \times 2$, with stride $2 \times 2 \times 2$, except for pool1, which has a kernel size of $1 \times 2 \times 2$ and stride $1 \times 2 \times 2$, with the intention of not to merge the temporal signal too early and to satisfy the clip length of 16 frames. Therefore, $7 \times 7 \times 512$ feature maps are generated, since we resize the input video frames to 120×120. According to the network configuration, as shown in Figure 3, we can completely collapse the temporal signal before concatenating with it the 2D features.

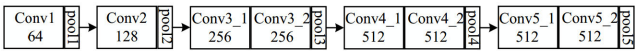

FIGURE 3. C3D deep structure.

B. REGRESSION-BASED TRACKING ARCHITECTURE

As demonstrated in Figure 2, the regression-based tracking architecture directly inherits the dual branches of our trained offline architecture, to form its backbone network. That means that the dual branches will not be updated again in the tracking procedure. After the backbone is followed by both the three fully connected layers ($fc1 - fc3$) in the upper path and the regressor network in the lower path, as illustrated in Figure 2. The fully connected layers and the regression network are trained discriminatively to differentiate the target from the background, and the offset of the bounding box is further regressed to finetune the candidate bounding boxes with the highest classification score. The three fully connected layers have the same hyper-parameter configuration as in the offline network. The regressor is composed of a fully connected layer, with four output units, representing the coordinates of the bounding-box center and the width and height of the bounding box, respectively.

C. HIERARCHICAL LOCAL-GLOBAL SPATIAL-TEMPORAL INTEGRATION

In the dual-branch architecture, the spatial appearance features and the temporal motion features are extracted and

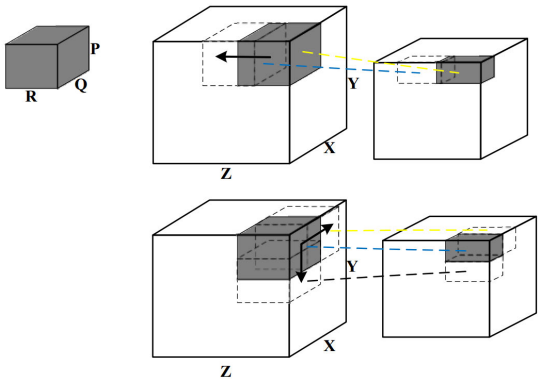

FIGURE 4. 3D convolution operations, where X, Y, and Z are the size of an input cube, and Q, P, and R are the convolutional kernel size. (Top: sliding along the temporal dimension, bottom: sliding over the spatial dimensions).

combined synchronously to achieve accurate object tracking. In the C3D branch, each 3D operation convolves a 3D kernel with the cubes formed by stacking multiple contiguous frames together. In this way, the pixels in the generated feature maps reflect the connections between the target appearances in the multiple contiguous frames, thereby capturing motion information. For clarity of presentation, Figure 4 illustrates the 3D convolution operation. From the figure, we can see that the R dimension of the 3D convolutional kernel represents the temporal dimension, which does not exist in a 2D kernel. Furthermore, the 3D convolution formulation can be written as:

$$v_{ij}^{xyz} = b_{ij} + \sum_{m} \sum_{r=0}^{R-1} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \quad (1)$$

where v_{ij}^{xyz} denotes the value of the unit at position (x, y, z)in the j_{th} feature map of the i_{th} layer. b_{ij} is the bias for this feature map, and m indexes over the set of feature maps from the $(i-1)_{st}$ to the current feature map. w_{ijm}^{pqr} is the $(p, q, r)_{th}$ value of the kernel connected to the m_{th} feature map of the previous layer. P and Q are the height and width of the kernel, respectively, and R is the temporal dimension of the 3D kernel. As demonstrated by Equation (1), the local neighborhood region centered at position (x, y) is clamped with the corresponding consecutive R frames in the computation of the spatial-temporal features. In other words, the motion of a tracked object in the R consecutive frames is embedded in the feature maps v, together with the appearance feature of the corresponding spatial patch (i.e. the (x, y)-centered neighboring region). Thus, the convolution operations locally synchronize the spatial-temporal information obtained from the target patch and those in the successive frames. Usually, the motion of a target patch may suffer from changes in its local appearance. Therefore, it is necessary to integrate the local spatial and the corresponding local temporal information synchronously. Furthermore, when a 3D convolution kernel slides along the time axis (as shown

at the top of Figure 4), we can obtain feature maps along the temporal dimension. The values of a feature map summarize the motion pattern in the neighborhood of the position (x, y). When the convolution kernel slides only along the x or y axis (as shown at the bottom of Figure 4), we can obtain a 2D feature map of values along the spatial dimensions. This set of values expresses the motion patterns of the different spatial blocks in the clamped consecutive frames. All the feature-map values are integrated to form the joint temporal and spatial features. If this spatial-temporal feature is not extracted in a synchronized manner, the misalignment of the spatial and temporal features can easily occur. This may hinder the capability of our tracker to handle various challenges in videos. Moreover, the extraction and fusion of the spatial and temporal features are performed by the convolutional kernels, whose parameters are learnable and flexible.

From the previous description, using the 3D convolution operations only, which focuses on the synchronous fusion of the local spatial-temporal information, is not sufficient to achieve the best tracking performance. In fact, the global motion pattern of a tracked target is exhibited in multiple successive video frames, and the appearance representation depends on the complete area covered by the target in the current frame, rather than the image patches only. Hence, we additionally incorporate the global integration of spatial and temporal information using a dual-branch architecture design. Specifically, the spatial and temporal information is fused by concatenating the 2D and 3D output features. Therein, feeding the current frame into the 2D network branch yields the appearance features, and feeding the frame cube, comprising of the current frame and a specified number of preceding frames, into the 3D network branch yields the temporal information. By sliding the window temporally, synchronized with the current frame, global spatial-temporal fusion occurs throughout the entire video.

In summary, the key to our dual-branch architecture is its ability to hierarchically fuse temporal-spatial information in a synchronous manner, using C3D for local integration of motion features, which are combined with the information about the tracked object obtained by 2D convolution for global integration. In addition, deep structures based on 2D and 3D convolutions can learn high-level target appearance and semantic-level motion features, which are then further fused. This fusion of semantic appearance and semantic motion, instead of combining low-level motion and high-level appearance as in [8], reflects the synergy in our proposed method. In our experiment, we will show that the proposed dual-branch model leads to consistent performance improvements.

D. TRAINING

Corresponding to the proposed dual-branch tracking structure, the training of the network consists of multidomain offline learning and online visual tracking, which will be described in the following subsections.

1) OFFLINE TRAINING

We train offline the multidomain architecture, in an attempt to integrate arbitrary domain information into the learning procedure. We believe that some common properties about both object appearance and object motion pattern exist in different domains, which are desirable for visual tracking [1]. The multidomain idea is borrowed from MDNet [1], but it just emphasizes the appearance learned from multiple domains, without any motion information being considered. The reason for this is that MDNet uses a deep 2D network only, instead of combining a 2D and a 3D network.

Specifically, in each SGD iteration, T frames are sampled randomly from each selected sequence for easy implementation. In a mini-batch for the C2D branch, 32 positive examples, i.e. intersection over union (IoU) > 0.7, and 96 negative examples (IoU < 0.5) were drawn from each frame, with a Gaussian distribution and a uniform distribution, respectively. This results in producing $32T$ positive and $96T$ negative samples for training, which are all from the same domain. For convenience, we denote each sample as $c^d_{t_i}$, representing the i_{th} sample from the t_{th} training frame of the d_{th} video sequence, where $d \in [1, D]$ and $i \in [1, N]$. N is the total number of samples from each frame, i.e., $N = 32{+}96 = 128$, and D is the number of video sequences used for training. Meanwhile, we construct a mini-batch for the C3D branch with the target-cube samples. Specifically, frame t and a specified number (e.g. 15 in this paper) of the preceding consecutive frames form a frame-cube. For each frame in the frame-cube, we crop a target-centered image region, which is twice the size of the ground-truth bounding box containing the targeted object, and then resize each of the regions in the respective frames into a fixed size, e.g. 120×120 pixels in our experimental setting. Finally, the cropped and resized regions form a sample point of the C3D mini-batch. This sample point is named as the target-cube, which is denoted as $[x^d_{t-15}, x^d_{t-14}, \ldots, x^d_{t-1}, \widetilde{x}^d_t]$. $\widetilde{x}^d_t$ is specially cropped, based on the center of the ground-truth bounding box of the preceding frame, with the size double that of the ground-truth bounding box. This setting is due to the fact that the ground-truth bounding box for the current frame t is unknown during tracking. It is reasonable to assume that the targeted object in the current frame is moving within the area that is twice the size of that in the previous frame. By packing the ground-truth regions, in terms of their pixel values, with double the size of their original ground truths, this can form contextual information about the targeted object in the successive frames, which can effectively represent the motion and appearance information about the object.

We denote *motion_info* and *appearance_info* as the output feature maps generated by the C3D and C2D branches, respectively. Furthermore, we denote $f_{t_i^d}$ as the concatenated feature maps output by the dual network. These two types of information can be expressed as follows:

$$motion\_info^d_{t_i} = C3D(x^d_{t_i-15}, x^d_{t_i-14}, \ldots, x^d_{t_i-1}, \widetilde{x}^d_{t_i}) \quad (2)$$

$$appearance\_info_{t_i}^{d} = C2D(c_{t_i^d}) \quad (3)$$

$$f_{t_i}^{d} = dual(c_{t_i^d}; x_{t_i-15}^{d}, x_{t_i-14}^{d}, \ldots, x_{t_i-1}^{d}, \widetilde{x}_{t_i}^{d}) = [motion\_info_{t_i}^{d}; appearance\_info_{t_i}^{d}] \quad (4)$$

where $C3D(\cdot)$ and $C2D(\cdot)$ represent the forward outputs of the C3D and the C2D branches, respectively. $dual\_net(\cdot)$ is the feature output of the backbone network, which is generated by synergistically extracting and fusing the motion and appearance information. Moreover, $f_{t_i}^{d}$ is fed forward into three fully connected layers, and finally a SoftMax function to determine whether the candidate $c_{t_i}^{d}$ is the target or the background. For the backpropagation adopted in SGD, we employ the cross-entropy loss function as follows:

$$\ell(y_{t_i}^{d}, s_{t_i}^{d}) = \sum_{c} y_{t_i}^{d} \times log(\sigma(s_{t_i}^{d})) \quad (5)$$

$$s_{t_i}^{d} = \phi^{d}(\Phi(f_{t_i}^{d})) \quad (6)$$

In (5) and (6), $\sigma(\cdot)$ denotes the output of the SoftMax classifier, $\Phi(\cdot)$ represents the output from the shared $fc1 - fc2$ layers, and $\phi^{d}(\cdot)$ is a binary classifier from the last fully connected layer $fc3^{d}$ for the domain d, and $s_{t_i^d}$ is the corresponding classification score. $y_{t_i} \in \{1, 0\}$ is a one-hot encoding of the ground-truth label, where 1 and 0 represent the target and the background classes, respectively. As stated previously, we chose $T = 8$ frames from a domain to form the mini-batch for each iteration. Therefore, the objective function to train the dual-synergy network offline for the d_{th} domain is thereby constructed as follows:

$$\arg\min_{W} \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{i=1}^{N} \ell(y_{t_i}^{d}, s_{t_i}^{d}) + \beta ||W||_2^2 \quad (7)$$

where W represents the weights of the dual network, and β indicates the weight decay.

2) ONLINE TRAINING

After the offline training, as described in Section D.1, the optimal parameters learned for the dual-branch backbone and the first two shared fully connected layers in the multidomain architecture are kept fixed, while the domain-specific layers, $fc3^{1} - fc3^{D}$, are fine-tuned to train up a new $fc3$ for each of the D training sequences. These optimal parameters are frozen and directly used in the tracking procedure, through which the common prior knowledge about object appearance and motion pattern can be applied to testing sequences, achieving the transfer of domain knowledge. This benefits the tracking of new targets, and alleviates the burden of online training. In fact, we simply fine-tune the last fully connected layers ($fc3^{1} - fc3^{D}$) in online training. The fine-tuning consists of two parts, tracker initialization and online update, details of which are given in the following.

a: TRACKER INITIALIZATION

We initialize our tracker with the ground-truth bounding box from the first frame; this is a standard practice used for object tracking. In detail, we collected 500 positive (IoU $>$ 0.7) and 5000 negative (IoU $<$ 0.5) samples to train the fully connected layers $fc1-fc3$ in Figure 2. In addition, we sampled the patches uniformly by setting the IoU in the range [0.6,1], and subsequently selected 100 patches, such that the ratio of the area of each of the patches to that of the ground-truth bounding box is limited to the range [1,2]. These samples were used to preliminarily learn the regressor model in the tracking architecture shown in Figure 2.

b: ONLINE UPDATE

Training Data Pool: At each subsequent frame t, the target position is estimated using the current tracker. Centered at the estimated target box, we collected 50 positive sample patches according to a Gaussian distribution, whose IoU with the predicted targeted bounding box is [0.6,1]. Similarly, we uniformly collected 100 negative samples, whose IoU with the predicted targeted bounding box is [0,0.3]. The IoU boundary values are more stringent compared to the sampling procedure for offline training, in consideration of the probable inaccuracy of the estimated bounding boxes. Finally, the estimated box, and all the positive and negative samples, together with the t_{th} frame, are included to form the training data pool. In consideration of the pool volume, we update the pool to the latest 100 frames of the associated data.

Regressor Enhancement: At frame P, we perform a one-time enhancement learning of the regressor. Specifically, we take all the estimated bounding boxes from frames 2 to P from the data pool, and then pad each box using the original image pixels to double its size. Subsequently, these padded boxes are resized into a fixed size of 120×120, and the target-cube is built using the ITT strategy, which will be described later. Furthermore, the target-cube is input into the C3D branch, capturing the motion features, to train the regressor under the supervision of the ground truth from the first frame. Compared to most of the existing regressors learned based on the appearance features [1], [24], our approach synergistically uses spatial-temporal information to learn the regression model, and thus renders it more accurate.

Periodical Updating: If t mod $P = 0$, we update the target/background classifier and its associated fully connected layers. From the data pool, we randomly select positive and negative samples to construct the mini-batch for SGD iteration, based on the binary cross entropy loss. This procedure is called the 'long-term update', as the samples may come from the frames up to 100 frames away. However, if all the candidates' scores are below the threshold '0' when predicting for frame t, we adopt the short-term update. This will only consider samples from the latest specified number of frames (16 in our experiments) used for the SGD iteration.

3) TRAINING STRATEGIES

a: INVERSE TEMPORAL TRAINING (ITT)

After the prediction for the P_{th} frame is completed, we use ITT to further enhance the learning of the regressor by

simulating the video replay process. Based on the target estimations from the first full cycle - i.e. the predictions from the first P frames - we reverse the order and then construct the input $[x_P, x_{P-1}, \ldots, x_2; x_1]$ for the C3D branch. Each $x_i (i = 2, 3, \ldots, P)$ is centered at the estimated target box in frame i, and we also double the size of the primary estimated box in the i_{th} frame to incorporate the contextual information. Unlike other frames that contain estimated bounding boxes and the corresponding centers smoothly transitioning backward, x_1 contains the ground-truth box, which could be very different from that of x_2. This greater center variation between x_1 and x_2 might disrupt the smooth replay. Therefore, x_1 is constructed by cropping the patch with the same center as that of x_2 and doubling its size as with the ground truth from the first frame. So far, just as tracking is started from the P_{th} frame and back to the first frame, all the x values are arranged in the temporally reverse order. We use the temporal-reversed cube $[x_P, x_{P-1}, \ldots, x_2; \widetilde{x}_1]$ as the input of the C3D branch, and the samples from the first frame, i.e. x_P, as the input of C2D, to retrain the regressor. To summarize, this new Inverse Temporal Training strategy helps us to introduce motion pattern to learn the regressor for improving localization precision, even in the case that we have the ground truth only in the first frame.

With the help of the ITT strategy, our regressor differs in two aspects from the existing regression techniques used in state-of-the-art trackers widely, including [1], [39], [41]. The first one is that our regression model exploits not only the ground-truth bounding box from the first frame, but also the estimated boxes in the subsequent frames within the first cycle (i.e., from the 2_{nd} to the 16_{th} frames in our setting). The second one is that the regressor is learned based not only the appearance information from the first frame, but also the motion information hidden in the first cycle of the frames. It can be deduced that these two characteristics make our regression model more accurate than the traditional ones [1]. The estimated boxes will not drift far away from the ground truth in the early tracking stage, and richer information is provided for training the model. Figure 5 gives a simple view of ITT, from which a similar motion pattern is implied in both the reverse sequence and the forward sequence, even though the trajectory renders with different directions. This claims the rationality of the ITT strategy.

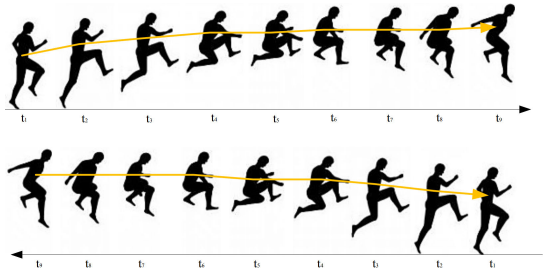

FIGURE 5. In the first row, video frames are arranged in the original, forward order. In the second row, video frames are arranged in the reverse order. The orange line represents the trajectory of the object.

b: FIRST STRATEGY

According to the data pool mechanism, as described previously, all the sample points are from the latest 100 frames. Thus, when we perform a periodical update at frame $t > 100$, no samples from the first frame are included in the data pool. Considering that the data points from the ground truth in the first frame are more reliable, we sample some positive data points from the first frame for SGD iterations. The introduction of FIRST slightly modifies the loss function as follows:

$$\sum_{k}\sum_{i} \ell(y_{t_i}, s_{k_i}) + \sum_{i+} \ell(y_{1_{i+}}, s_{1_{i+}}) + \beta||W|| \qquad (8)$$

where k denotes a random frame index number in the latest 100 frame range, i denotes the sample index from the data pool, and $i+$ denotes the positive sample index from the first frame.

IV. EXPERIMENTS

In this section, we first describe the experiment settings. Then, the ablation studies are provided to analyze the effect of the components or strategies in the proposed method. Finally, quantitative and qualitative results on the tracking benchmarks, including OTB series datasets (OTB2013, OTB2015, OTB50), UAV123 and TC128, and VOT challenge datasets (VOT2015, VOT2016) are presented, for comparing our method with those state-of-the-art tracking algorithms.

A. EXPERIMENT SETTING

All the experiments were implemented under the Pytorch framework, on a PC equipped with a single Intel(R) Xeon (R) E5-2630 CPU @ 2.4GHz and a single TITAN X GPU with 12G RAM. The running speed of our MADual tracker is 0.7 fps. We applied the SGD solver to perform offline training of the multidomain architecture, which converges after 25 epochs. The learning rate of the dual-branch backbone, consisting of C3D and C2D, is set at 10^{-3}, and set at 10^{-2} for the training of the subsequent fully connected layers. In addition, we freeze the parameters of the first two layers of the C3D convolutional network during online training, in order to achieve a higher training efficiency. For online tracking, we trained the network for 30 epochs, with the learning rate of 10^{-4}, based on the first frame of the test video. For online updating, the number of epochs for fine-tuning is 15. The weight decay and momentum are set at $5 \times 10^{-3}$and 0.9, respectively.

B. ABLATION STUDY

To demonstrate the effectiveness of the proposed method (including the hierarchical local-global motion-appearance integration architecture, as well as the ITT and FIRST strategies), we conducted comparison experiments between the proposed MADual and three variants of MADual, which are as follows:

(1) Single_2D. This is obtained by removing the C3D branch from the deep dual-branch backbone network. From

the architectural point of view, Single_2D is the same as MDNet. However, we keep the optimal parameters and hyper-parameters for the MADual network, and evaluate the impact on performance when the C3D branch is removed. The ITT and FIRST strategies are not used in this Single-2D.

(2) Dual_Preliminary. This is obtained by using both the C2D and the C3D branches to form the dual-branch backbone, without using any of the two strategies.

(3) Dual_ITT. This is obtained by equipping Dual-Preliminary with the ITT strategy.

The comparison experiments were conducted on OTB2013 [11]. Table 2 summarizes all the results, in terms of overlap precision (OP) and distance precision (DP) (described later). From Table 2, MADual achieves the best accuracy in terms of DP and OP, i.e. up to 94.6% and 72.5%, respectively. Using Single_2D decreases the tracking performance significantly, with the DP and OP reduced to 91.5% and 69.8%, respectively. In addition, whether introducing C3D, or utilizing the ITT or FIRST strategies, can help improve the performance compared to the basic Single_2D method. Therefore, this proves the rationality of the design of our proposed method. When looking closer at the results, Dual_Preliminary significantly outperforms Single_2D by 1.6%, in terms of both DP and OP. This clearly shows that the hierarchical local-global appearance-motion integration mechanism is advantageous to visual tracking. With the help of ITT, Dual_ITT achieves a high DP and OP of 94.2% and 72.1%, respectively. Including the FIRST strategy can further improve the performance of Dual_ITT.

TABLE 2. Ablation study of the different components of the proposed MADual tracker on OTB2013.

Operation	3D	ITT	FIRST	OP(%)	DP(%)
Single_2D				91.5	69.8
Dual_Preliminary	✓			93.1	71.4
Dual_ITT	✓	✓		94.2	72.1
MADual	✓	✓	✓	94.6	72.5

C. EVALUATION ON THE OTB SERIES DATASETS

The OTB series datasets include OTB2013, OTB2015 and TB50. Herein, the OTB2013 and OTB2015 benchmarks contain 51 videos and 100 videos, respectively. The latter benchmark is an extension of the former dataset. Within these two datasets, 50 particularly challenging sequences were selected to build the TB50 dataset.

For this part of the evaluation, we selected 58 videos from the VOT challenge datasets, VOT2013, 2014 and 2016, (as described later), excluding the videos in the OTB series, UAV123 and TC-128 (as described later) to train the proposed dual-branch based tracker offline.

We use two metrics: success rate and precision, for quantitative evaluation. The success rate is computed as the ratio of successfully tracked frames according to the IoU of the predicted and ground-truth bounding boxes, and the success plot shows how the success rate changes with the overlap thresholds varying from 0 to 1. Precision is defined based on the center localization error, and the precision plot is plotted to show the changes of the ratio of the frames with a localization error below a threshold. Furthermore, OP is defined as using the success rate at a specific threshold (e.g. 0.5) for ranking the trackers' performance. DP is defined as the precision value at a specific threshold (e.g. 20 pixels). The state-of-the-art trackers, MDNet [1], ECO [20], C-COT [19], DeepSRDCF [4], SRDCF [21], ADNet [57], PTAV [17], CFNet [58], SiamRPNRes22 [28], SiamFC-Next22 [28], DAT [59] and Staple [16], RFL [42], were used for comparison. The experimental results are demonstrated in Figures 6 to 8, in terms of the precision and success plots. Note that OPE in these figures refers to the one-pass evaluation, as was explained in [11]. All the trackers are ranked according to DP and OP.

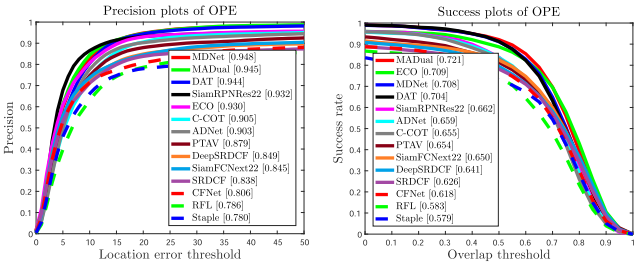

FIGURE 6. Precision and success plots on the OTB2013 dataset.

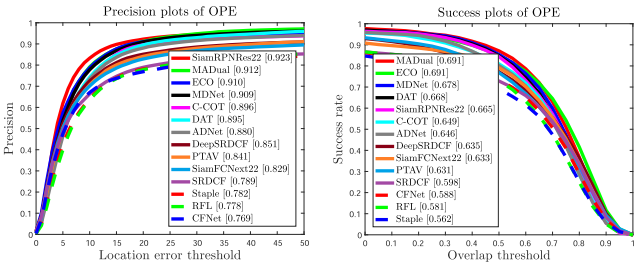

FIGURE 7. Precision and success plots on the OTB2015 dataset.

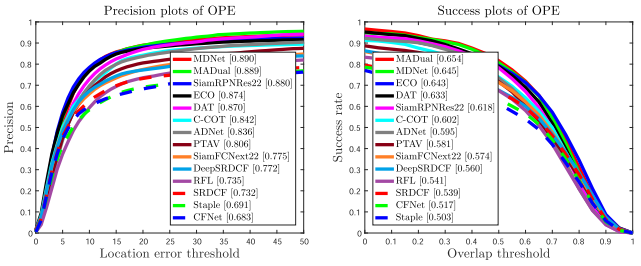

FIGURE 8. Precision and success plots on the TB50 dataset.

Overall, the performance of our MADual tracking method is ranked first or at a high position for all the OTB datasets, in terms of all the evaluation metrics. Specifically, the MADual tracker achieves the highest OP of 72.5% and 65.4% on OTB2013 and TB50, respectively. Compared with the second-ranked methods, such as ECO and MDNet, ours achieves an increase of the success rate by 1.6% and 0.9%,

respectively. On the largest OTB 2015 dataset, both our MADual and ECO achieve the best performance in terms of OP. On the other hand, our method is ranked second on the OTB series datasets, in terms of DP, and is only 0.2% and 0.7% lower than MDNet on OTB2013 and OTB2015, respectively. In addition, we found that the green curves, which represent the DP performance of our MADual method in Figures 6, 7 and 8, always remain in the outer area for thresholds larger than 30, indicating that our distance precision rates are higher than the methods ranked first. Table 3 tabulates the detailed precision value of several top performing methods (MADual, ECO, SiamRPNRes22 and MDNet) with different location-error thresholds, which further proves the superior performance of our proposed tracker.

TABLE 3. Precision value at different pixel distances. The results are shown in the form of "OTB2013/OTB2015/TB50". The top two methods are highlighted in red and blue, respectively.

Tracker	DP@30(%)	DP@40(%)	DP@50(%)
MADual	97.4/92.8/94.8	98.4/94.7/96.4	98.7/95.9/97.1
ECO	94.5/89.7/93.4	95.3/90.9/94.5	95.7/91.6/95.0
MDNet	97.3/91.7/94.0	98.1/93.1/95.4	98.2/93.9/96.1
SiamRPNRes22	94.4/90.5/94.5	95.1/91.8/95.5	95.4/92.4/95.1

We further investigated the capability of the MADual tracker to handle various challenging conditions. As we know, Wu et al. [12] categorized the OTB2015 sequences according to 11 attributes and constructed several subsets with specific challenging conditions. We analyzed the tracking performance of the MADual method under different challenging conditions. Figure 9 shows the bar charts, in terms of OP and DP, respectively. We can see that MADual is always ranked within the top three, in terms of both OP and DP. This indicates that the proposed tracker generally performs well in dealing with different challenging conditions.

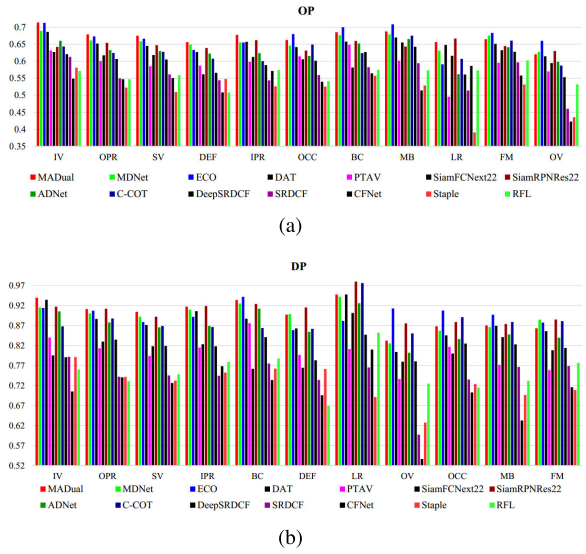

FIGURE 9. The performance, in terms of (a) OP and (b) DP, of 13 trackers under different challenging conditions on OTB2015.

Specifically, in terms of OP, MADual performs favorably against all of the state-of-the-art methods in five challenging cases, including deformation (DEF), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), and illumination variation (IV). As for the cases of occlusion (OCC), background clutters (BC) and motion blur (MB), our MADual still outperforms MDNet, but is in second position, in terms of the success rate, just behind the ECO method. As shown in Figure 9(b), with IPR, OPR, SV and IV challenges, MADual outperforms all the other trackers in terms of distance precision. In the cases of the DEF and BC challenges, our method outperforms MDNet, and is ranked second. In the cases of OCC, OV, LR and MB, the performance of the MADual tracker takes third position. To summarize, our MADual achieves the best performance in most of the difficult scenarios.

D. EVALUATION ON TC-128 AND UAV123

The UAV123 dataset consists of 123 aerial videos, with more than 110k frames. The TC-128 dataset contains 128 color videos. We used the same metrics and training dataset as those used on OTB in our experiments. Those state-of-the-art trackers that provide publicly-available results on TC-128 or UAV123 are used for comparison. Specifically, ECO [20], RTMDNet [41], PTAV [17], DeepSRDCF [4], SRDCF [21], and KCF [31] were evaluated on TC-128. ECO, RTMDNet, SRDCF and KCF were evaluated on UAV123.

Figure 10 illustrates the precision and success plots of the proposed MADual tracker and all the other algorithms evaluated on TC-128. It can be seen that our method is ranked first in terms of DP, in particular, it outperforms the state-of-the-art method ECO by 1%. However, our MADual achieves the success plot of 59.5%, which is slightly lower than ECO. Figure 11 shows the precision and success plots of the different methods on the UAV123 dataset. The results show that our MADual tracker is obviously superior to ECO, in terms of the distance precision and overlap success rate. In detail, our method outperforms ECO by 1.0% and 2.4%, in terms of DP and OP, respectively.

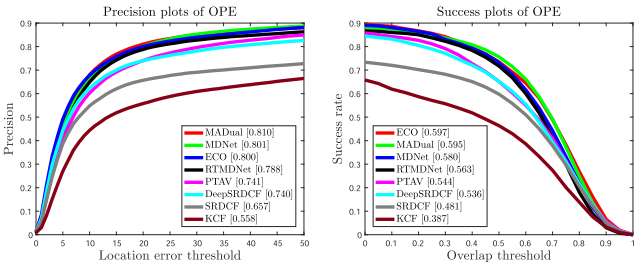

FIGURE 10. Precision and success plots on theTC-128 dataset.

E. EXPERIMENT ON VOT CHALLENGES

In this part of evaluation, we use 87 training sequences from OTB2015, excluding the videos included in VOT2015 or VOT2016, for offline training.

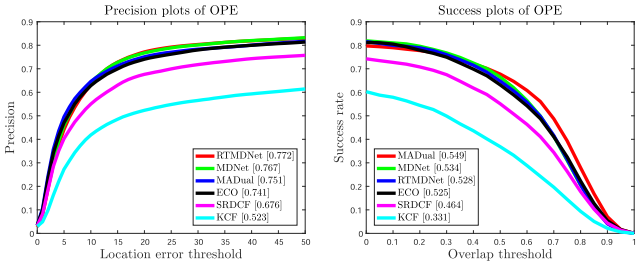

FIGURE 11. Precision and success plots on the UAV123 dataset.

We use the official version 6.0.3 of the Visual Object Tracking toolkit to compute and plot the reset-based performance measures, including accuracy, robustness and expected average overlap. Herein, the accuracy measures how well the bounding box predicted by a tracker overlaps with the ground-truth bounding box in terms of IoU. The robustness measures the frequency of tracking failures. The expected average overlap (EAO) measures the overall performance, which takes both accuracy and robustness into account quantitatively. The detailed formulation of these metrics can be found in [60].

1) VOT2015 CHALLENGE

The VOT2015 dataset consists of 60 sequences. We compare the proposed MADual tracker with the top 5 trackers in the VOT2015 challenge [15]. Likewise, we compare our tracker with several recently proposed algorithms, including HCFTs [18], TADT [61] and three Siamese network-based trackers (SiamRPN [3], SiamFCNext22 [28] and SiamRPNRes22 [28]), as their results were reported, and they represent the state-of-the-art tracking algorithms. All the results are summarized in Figure 12 and Table 4. From Figure 12, our MADual achieves the best result, significantly better than SiamRPNRess22 [28], which is ranked second. The results in Table 4 show that our MADual outperforms SiamRPNRes22 and MDNet by 7.1% and 7.9%, respectively, in terms of EAO. Although the performance of our proposed tracker is slightly lower than MDNet in terms of accuracy (A), it achieves better performance in terms of robustness (R).

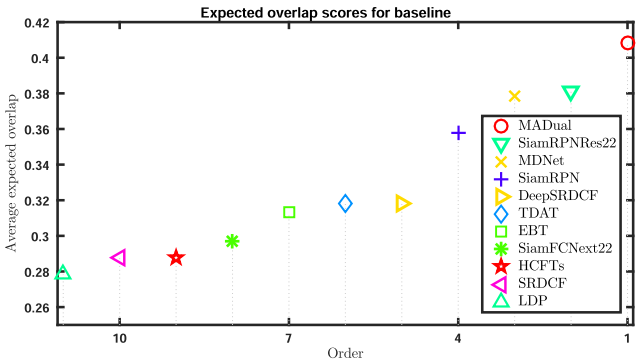

FIGURE 12. Expected average overlap (EAO) plot for VOT2015.

TABLE 4. The comparison of the proposed MADual with the top 5 trackers in the VOT2015 challenge and some recently proposed state-of-the-art methods, in terms of overlap, failures, and expected average overlap (EAO), using the VOT toolkit (version 6.0.3).

Tracker	A	R	EAO
MDNet	0.595	0.24	0.378
DeepSRDCF	0.564	0.32	0.318
EBT	0.453	0.29	0.313
SRDCF	0.552	0.38	0.287
LDP	0.485	0.44	0.279
HCFTs	0.485	0.30	0.288
SiamRPN	0.602	0.29	0.358
TADT	0.579	0.35	0.318
SiamFCNext22	0.56	0.46	0.297
SiamRPNRes22	0.579	0.24	0.381
MADual	0.59	0.22	0.408

2) VOT2016 CHALLENGE

The video sequences in VOT2016 are the same as those in VOT2015, while the ground-truth bounding boxes are precisely re-annotated. We compare our tracker against the top five trackers, i.e. C-COT [19], TCNN [62], SSAT [6], MLDF [6] and Staple [16], in the VOT2016 challenge, as well as three recent state-of-the-art trackers, TADT [61], SiamFCNext22 [28], and SiamRPNRes22 [28]. In addition, ECO and MDNet are selected for comparison. ECO is considered because of its excellent performance on the OTB series datasets, and MDNet, because of its correlation to the proposed MADual. The top three trackers, in terms of the EAO ranking plots, are SiamRPNRes22, MADual and ECO, which have similar performance, as shown in Figure 13. Table 5 shows the detailed performance of the different methods in terms of A, R and EAO. We also summarize the no-reset AO measures, which was officially added in the VOT2016 competition evaluation. As shown in Table 5, our tracker is ranked first in terms of robustness and EAO, and ranked second in terms of the AO metric.

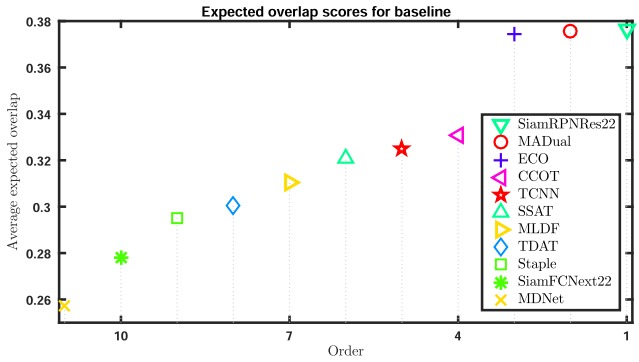

FIGURE 13. Expected average overlap (EAO) plot for VOT2016.

As described in [6], all the sequences in the VOT2016 challenge are per-frame annotated with visual attributes, such as camera motion, illumination change, occlusion, size change, motion change, etc., which correspond to the most difficult tracking tasks. Based on the attribute annotations, we further investigate the ability of the MADual tracker to handle chal-

TABLE 5. Comparison of the proposed MADual with the top five trackers in the VOT2016 challenge and five state-of-the-art methods proposed in recent two years, in terms of overlap, failures, expected average overlap (EAO), with the use of the VOT toolkit (version 6.0.3).

Tracker	A	R	AO	EAO
C-COT	0.558	0.320	0.470	0.331
TCNN	0.551	0.360	0.487	0.325
SSAT	0.570	0.410	0.516	0.321
MLDF	0.511	0.310	0.423	0.311
Staple	0.555	0.550	0.390	0.295
MDNet	0.547	0.380	0.458	0.257
ECO	0.576	0.270	0.441	0.374
TADT	0.569	0.470	-	0.301
SiamFCNext22	0.557	0.510	-	0.278
SiamRPNRes22	0.575	0.330	-	0.376
MADual	0.555	0.230	0.515	0.376

lenging tracking scenarios. Table 6 summarizes the robustness, accuracy and EAO of the different trackers, suffering from different visual attributes or challenges. From the table, different trackers perform differently in terms of accuracy, robustness and EAO over the different attributes. The top three best-performing trackers, in terms of accuracy, are SSAT, SiamRPNRes22, and our MADual. However, in terms of EAO, ECO takes the place of SSAT to squeeze into the top three performers. Therefore, SiamRPNRes22 and our MADual perform very well, in terms of both accuracy and EAO. However, SiamRPNRes22 falls behind the top three, while our MADual is significantly better than all the other state-of-the-art trackers in terms of robustness. In summary, our MADual is always ranked in the top three, which implies that it performs very well in terms of accuracy, robustness, and EAO. In other words, our MADual performs well in diverse visual attributes, which shows its power to handle different challenging sequences.

In detail, our MADual achieves the best performance in terms of accuracy on the size-change attribute, i.e. A =0.528, which is 2.3% higher than that of SiamRPNRes22, which is ranked second. Under both camera motion and occlusion, the MADual tracker achieves the third place in terms of the same metric, and MADual is only 0.7% lower than that of the SiamRPNRes22, which is ranked first on the occlusion attribute. We notice that SSAT achieves the first place in terms of accuracy among all the other listed trackers, with respect to all the visual attributes. The reason for this is that SSAT concentrates on estimating the tightest bounding box of targets, and is an extension of the MDNet framework. To this end, SSAT introduced a segmentation technique, and further inferred whether or not a target was occluded. When the target is occluded, training examples from that frame are not extracted for updating the tracker. However, in terms of robustness or EAO, SSAT falls behind the top three best-performing trackers.

In terms of robustness, our MADual performs favorably against all the evaluated trackers on all the visual attributes, as shown from the R table in Table 6. This implies that our MADual fails the least over all the different visual attributes. In detail, MADual achieves significantly better performance in terms of robustness, i.e. R = 0 and 0.285 on the illumination change and occlusion attributes, respectively, and they achieve a significantly lower robustness score, compared to that of the second best tracker ECO. Regarding the other three visual attributes, i.e. motion change, size change, and camera motion, MADual achieves second place, with results of 0.182, 0.546 and 0.113, respectively.

From the performance of EAO shown in Table 6, MADual performs slightly worse than SiamRPNRes22, in general. To be more specific, our tracker achieves the best performance, and EAO, being the second best, has its performance 21.2% higher than SiamRPNRes22 on the motion-change attribute. On occlusion and camera motion, our MADual is ranked second in terms of EAO, which are 0.306 and 0.385, respectively, and outperforms ECO and SiamRPNRes22 by 8.5% and 4.9%, respectively, on the occlusion attribute. In addition, our method is ranked third, behind SiamRPNRes22 and ECO, with the EAO value of 0.384 on the size-change attribute. Only on the attribute of illumination change, our MADual is ranked fourth. The reason for this may be that the corresponding accuracy is low, which drag down the EAO performance.

In summary, our MADual is the only one that is ranked the top three in terms of all the three metrices: A, R and EAO, so achieves the best overall performance. In particular, our MADual is prominent in the robustness performance on all the challenging conditions, ranked first for all the visual attributes. The reason for MADual achieving this performance can be deduced from the following facts. A target object is associated with a certain motion pattern. Even if the object suffers from appearance variation due to complex conditions, such as deformation, occlusion, rotation, size changing, as well as camera motion and motion change, the motion features obtained by C3D can maintain the connection of the target object in the different frames. With the help of the motion features, the MADual tracker can still locate the target under severe conditions. The complex scenarios that appear in some of the frames can be regarded as the noise of the regular motion pattern of the object being tracked in a local temporal region. As the motion pattern described by C3D is based on multiple consecutive frames (e.g. 16 in this paper), the influence of these noises on the motion features is limited. On the contrary, relying only on the appearance features obtained by C2D to locate the target object in the current frame, the tracking may easily fail in two cases. One case is that a very similar object distractor appears, such that the target object cannot be distinguished from the similar object. The other case is that the target object changes greatly and becomes significantly different from its appearance model. Different from the previous methods [1], which predict the location of the same target in the subsequent frames simply by using an appearance model, they may be affected by large appearance variations or may be distracted by a similar semantic object, so they perform poorly.

TABLE 6. The overall accuracy (A), robustness (R) and average overlap (EAO) averaged over the attributes on the VOT2016 dataset. The attributes tested include camera motion (Cam), illumination change (Illu), motion change (Mot), occlusion (Occ), and size change (Size). Red, blue and green fonts indicate the top-3 trackers, respectively.

A	C-COT	TCNN	SSAT	MLDF	Staple	MDNet	ECO	TADT	SiamFCNext22	SiamRPNRes22	MADual
Cam	0.578	0.581	0.613	0.523	0.583	0.563	0.595	0.581	0.565	0.622	0.599
Illu	0.656	0.642	0.668	0.58	0.715	0.639	0.646	0.628	0.674	0.563	0.632
Occ	0.475	0.526	0.536	0.455	0.511	0.508	0.5	0.527	0.509	0.539	0.532
Size	0.437	0.508	0.508	0.411	0.437	0.491	0.408	0.457	0.475	0.516	0.528
Mot	0.507	0.513	0.555	0.445	0.516	0.511	0.519	0.502	0.495	0.515	0.504
R	**C-COT**	**TCNN**	**SSAT**	**MLDF**	**Staple**	**MDNet**	**ECO**	**TADT**	**SiamFCNext22**	**SiamRPNRes22**	**MADual**
Cam	0.155	0.119	0.125	0.155	0.183	0.26	0.070	0.141	0.141	0.155	0.113
Illu	0.201	0.315	0.228	0.201	0.704	0.382	0.101	0.503	0.402	0.101	0
Occ	0.407	0.45	0.442	0.386	0.712	0.435	0.346	0.651	0.57	0.488	0.285
Size	0.588	0.644	0.994	0.714	1.008	0.541	0.714	0.924	1.092	0.714	0.546
Mot	0.263	0.302	0.304	0.141	0.303	0.244	0.202	0.323	0.404	0.222	0.182
EAO	**C-COT**	**TCNN**	**SSAT**	**MLDF**	**Staple**	**MDNet**	**ECO**	**TADT**	**SiamFCNext22**	**SiamRPNRes22**	**MADual**
Cam	0.354	0.319	0.315	0.306	0.318	0.252	0.42	0.353	0.272	0.367	0.385
Illu	0.402	0.331	0.387	0.443	0.269	0.313	0.648	0.261	0.29	0.469	0.419
Occ	0.246	0.258	0.181	0.204	0.235	0.218	0.282	0.251	0.188	0.322	0.306
Size	0.327	0.314	0.35	0.352	0.312	0.312	0.387	0.298	0.286	0.404	0.384
Mot	0.249	0.275	0.311	0.26	0.221	0.238	0.315	0.23	0.228	0.321	0.389

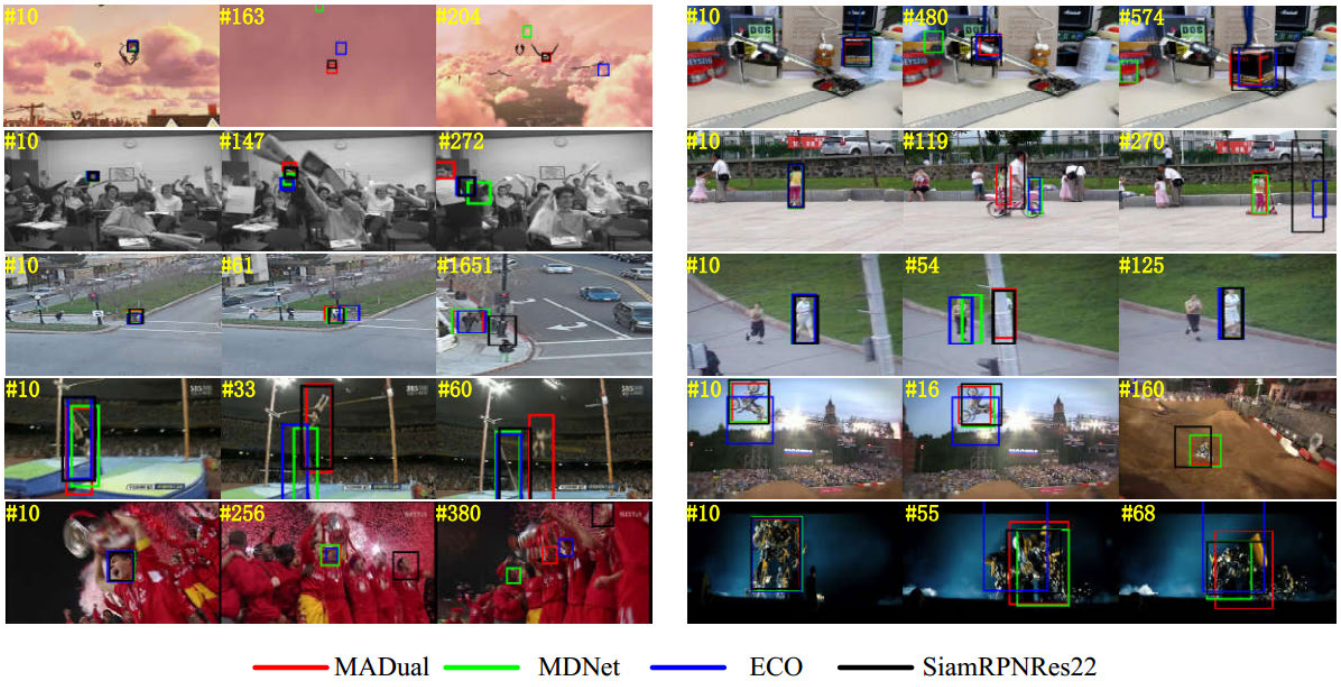

FIGURE 14. Qualitative evaluation of MADual, MDNet, ECO and SiamRPNRes22 on 10 challenging sequences (from left to right and up to down: Bird1, Box, Freeman4, Girl2, Human3, Jogging2, Jump, MotorRolling, Soccer, Trans) from OTB2015.

F. QUALITATIVE EVALUATION

Figure 14 provides the visual comparison of the best performed trackers, including SiamRPNRess22 [28], MDNet [1], ECO [20], and our MADual, on 10 challenging sequences from OTB2015. Overall, our MADual is able to locate targets accurately in complicated scenes.

In the 'Bird1' sequence, the target undergoes the out-of-view challenge, as shown in the 163rd frame. From frame 204, all the methods lose their tracking of the target object, while the proposed MADual tracker can recover its tracking. In the 'Girl2' sequence, the target is occluded, and suffers from deformation and scale variations. Our MADual and the MDNet methods perform tracking with satisfactory performance, while ECO and SiamRPNRess22 completely drift away from the target. The similar challenging scenario occur in the sequence 'human3', and our MADual tracker can still track the changing object successfully. In the 'Jump' sequences, the object undergoes severe deformation and rotation. All the methods, except the proposed MADual tracker, locate the object wrongly. On the contrary, our tracking algorithm estimates the bounding box precisely. In these sequences, the target objects exhibit heavy occlusion, motion

blur deformation, scale variations, and rotation and background clutter. For clarity and better presentation, only the results from the top three trackers on OTB2015: i.e. SiamRPNRess22, MDNet, and ECO, and our MADual, are shown. Overall, our MADual is able to locate the targets well in complicated scenes.

V. CONCLUSION

In this paper, we propose a new parallel network structure for tracking, namely MADual, which can effectively integrate the spatial-temporal information in a collaborative way, so as to improve tracking performance. In this method, the convolutional 3D (C3D) network is introduced to extract the internal spatial and temporal information, which is integrated to produce semantic motion pattern features. The convolutional 2D (C2D) network is employed as another branch to extract the appearance features. We combine the outputs of C3D and C2D to build up a dual-branch architecture, fulfilling the external spatial-temporal synergy. With the use of the Inverse Temporal Training and FIRST strategies, the proposed MADual framework is trained to achieve efficient object tracking. Extensive experiments were conducted on the OTB series dataset, the TC-128 dataset, the UAV123 dataset, and the VOT datasets. The experiment results show that the proposed method achieves a highly promising tracking performance, and is especially good at handling challenging conditions, such as deformation, scale variation, illumination changes, etc.

In our future work, we will formulate the proposed idea of hierarchical spatial-temporal structure into a real-time framework, such as SiamFC. This is expected to improve the tracking efficiency, as the current architecture is induced from the non-real-time MDNet tracker.

REFERENCES

[1] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.
[2] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang, "CREST: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2555–2564.
[3] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
[4] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCV)*, Dec. 2015, pp. 621–629.
[5] L. Bertinetto, J. Valmadre, F. J. Henriques, A. Vedaldi, and H. S. P. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.
[6] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, G. Fernandezac, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2016, pp. 777–823.
[7] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1449–1458.
[8] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 548–557.
[9] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 1243–1248.
[10] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1420–1429.
[11] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
[12] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[13] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 445–461.
[14] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.
[15] M. Kristan, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 1–23.
[16] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.
[17] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5487–5495.
[18] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.
[19] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.
[20] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939.
[21] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[23] Z. He, Y. Fan, J. Zhuang, D. Yuan, and H. L. Bai, "Correlation filters with weighted convolution responses," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2017, pp. 1992–2000.
[24] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 749–765.
[25] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4834–4843.
[26] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1763–1771.
[27] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional siamese network for high performance online visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4854–4863.
[28] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4591–4600.
[29] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1822–1829.
[30] S. E. Kahou, V. Michalski, R. Memisevic, C. Pal, and P. Vincent, "RATM: Recurrent attentive tracking model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1613–1622.
[31] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[32] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[33] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.
[34] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1186–1197, Jul. 2008.

[35] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.
[36] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[37] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, no. 5, p. 6, 2006.
[38] L. Zhang, J. Varadarajan, P. N. Suganthan, N. Ahuja, and P. Moulin, "Robust visual tracking using oblique random forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5589–5598.
[39] B. Han, J. Sim, and H. Adam, "Branchout: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3356–3365.
[40] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.
[41] I. Jung, J. Son, M. Baek, and B. Han, "Real-time mdnet," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 83–98.
[42] T. Yang and A. B. Chan, "Recurrent filter learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2010–2019.
[43] A. Kosiorek, A. Bewley, and I. Posner, "Hierarchical attentive recurrent tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3053–3061.
[44] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
[45] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[46] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4507–4515.
[47] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—Video to text," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4534–4542.
[48] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5734–5743.
[49] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1049–1058.
[50] H. Xu, A. Das, and K. Saenko, "R-C3D: Region convolutional 3D network for temporal activity detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5783–5792.
[51] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.
[52] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014, *arXiv:1405.3531*. [Online]. Available: https://arxiv.org/abs/1405.3531
[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[54] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Deep end2end voxel2voxel prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2016, pp. 17–24.
[55] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4193–4202.
[56] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[57] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1349–1358.
[58] J. Valmadre, L. Bertinetto, J. A. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2805–2813.
[59] S. Pu, Y. Song, C. Ma, H. Zhang, and M.-H. Yang, "Deep attentive tracking via reciprocative learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1931–1941.
[60] M. Kristan, "The visual object tracking VOT2013 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 1–23.
[61] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2019, pp. 1369–1378.
[62] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, *arXiv:1608.07242*. [Online]. Available: https://arxiv.org/abs/1608.07242

HAOJIE LI received the B.S. degree in electronics and information engineering from Shantou University, where he is currently pursuing the master's degree in electronic and communication engineering with the South China University of Technology, Guangzhou, China. His research interests include machine learning, computer vision, and visual tracking.

SIHANG WU received the B.S. degree from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the master's degree in information and communication engineering. His current research interests include machine learning, deep learning, reinforcement learning, and scene text detection.

SHUANGPING HUANG received the B.S. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 1995, and the M.S. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2005 and 2011, respectively. She is currently a Professor with the College of Electronic and Information Engineering, South China University of Technology. Her research interests include scene text detection and recognition in image and video, image processing, machine learning, and intelligent systems.

KIN-MAN LAM received the bachelor's degree in electronic engineering from The Hong Kong Polytechnic University, Hong Kong, in 1986, the M.Sc. degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology and Medicine, London, U.K., in 1987, and the Ph.D. degree from the Department of Electrical Engineering, University of Sydney, Sydney, NSW, Australia, in 1996. He was a Lecturer with the Department of Electronics Engineering, The Hong Kong Polytechnic University, from 1990 to 1993. He joined the Department of Electronics and Information Engineering, The Hong Kong Polytechnic University, as an Assistant Professor, in 1996, where he became an Associate Professor, in 1999. He was involved in professional activities. He is currently a Professor with The Hong Kong Polytechnic University. His research interests include human face recognition, image and video processing, and computer vision.

He was a member of the Organizing Committee or Program Committee of many international conferences. He was a Secretary of the IEEE International Conference on Acoustics, Speech, and Signal Processing, in 2003, the Technical Chair of the International Symposium on Intelligent Multimedia, Video and Speech Processing, in 2004, the Technical Co-Chair of the International Symposium on Intelligent Signal Processing and Communication Systems, in 2005, and the Pacific-Rim Conference on Multimedia, in 2010, a Secretary of the International Conference on Image Processing, in 2010, and the General Co-Chair of the IEEE International Conference on Signal Processing, Communications and Computing in Hong Kong, in 2012. He was the Chairman of the IEEE Hong Kong Chapter of Signal Processing, from 2006 to 2008. He has served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the Director of Student Services of the IEEE Signal Processing Society. He is the Vice President of Member Relations and Development of the Asia–Pacific Signal and Information Processing Association (APSIPA) and the Director of Membership Services of the IEEE Signal Processing Society. He is an Area Editor of IEEE *Signal Processing Magazine* and an Associate Editor of the *Digital Signal Processing*, APSIPA *Transactions on Signal and Information Processing*, and EURASIP *International Journal on Image and Video Processing*. He is also an Editor of HKIE *Transactions*.

XIAOFEN XING received the M.S. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2004 and 2013, respectively. She is currently an Associate Professor with the College of Electronic and Information Engineering, South China University of Technology. Her research interests include computer vision, affective computing, machine learning, and intelligent systems.