Triplet Network with Multi-level Feature Fusion for Object Tracking

Yang Cao, Bo Wan, Quan Wang, Fei Cheng
School of Computer Science and Technology, Xidian University, Xian, Shaanxi, China
Email: wanbo@xidian.edu.cn

***Abstract*—In recent years, Siamese network-based trackers have received increasing interest because of the balanced accuracy and speed. However, these tracking methods only extract the high-level features as target representations and merely utilize the first frame as the exemplar branch, which are less discriminative to distinguish similar distractors and are vulnerable to background clutter. To address these issues, we propose a novel Triplet network with a multi-level feature fusion structure (TripMFF) to leverage the merits of Triplet network and fuse multi-layer features for robust object tracking. Firstly, the Triplet network is adopted as our backbone architecture to take full advantage of the correlation among frames in the video sequence and make the exemplar branches have more useful information. Secondly, in order to capture more abundant features, a multi-level feature fusion structure is put forward to combine the low-level fine-grained and high-level abstract information, which improves the discriminative capability and stability of the tracker. Experimental results on tracking benchmarks prove that our proposed method achieves competitive performance and real-time tracking speed compared with other state-of-the-art trackers.**

***Contribution*—We propose a tracker TripMFF which adopts the Triplet network as backbone architecture and employs a multi-level feature fusion structure for robust object tracking.**

***Keywords*—Siamese network, Triplet network, object tracking, multi-level feature fusion**

I. Introduction

Object tracking is one of the most important problems in computer vision and has been widely used in various fields [1], [2], including video surveillance, virtual reality, robotics, and autonomous driving. In general, its task is to predict the position, size, and trajectory of the target in a sequence of images given the initial appearance and position (a bounding box) of an arbitrary object in the first frame. Despite remarkable progress in recent decades, finding the location of the same target in different frames is still a challenging issue because of many factors such as occlusion, rotation, low resolution, and motion blur. In this case, with the higher demands of accuracy and real-time performance, it requires trackers to be robust and efficient.

Recently, with the rapid development of convolutional neural networks (CNNs), deep learning-based tracking methods have shown great potential in tracking accurately and efficiently. Current works in these trackers could be roughly divided into two groups. The first group [4]–[6] boosts the tracking performance by adopting an online update strategy, where they update the parameters every few frames to adapt to the variations in target appearances. However, it is hard to meet the real-time requirement due to the large number of features and high computational load. Trackers [3], [7] based on the Siamese network are representative in the second group and have attracted much attention in the tracking community recently. These Siamese trackers transfer object tracking into a similarity learning problem and rely on offline training, which enables them to obtain competitive performance and high tracking speed. Among these methods, Luca et al. [3] propose a tracking algorithm named SiamFC by introducing a Siamese network which is fully-convolutional and trained offline to implement accurate and efficient tracking. Although this approach has achieved outstanding performance and fast speed as well as can handle occlusion and scale variation, the tracker is easy to draft and fails to track successfully when there are many similar distractors in the background as demonstrated in Fig. 1. Consequently, its tracking accuracy still has an obvious gap compared with other top-ranking

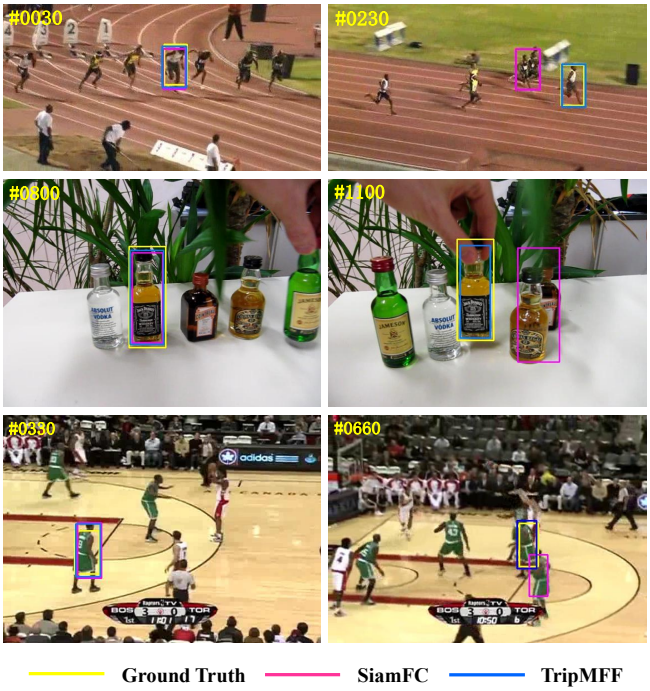


Fig. 1. Comparisons between SiamFC [3] and TripMFF on three challenging sequences: Bolt2, Liquor, and Basketball. Due to the Triplet network and multi-level feature fusion architecture, TripMFF is able to distinguish the target from distractors, while SiamFC drifts to the background.

978-1-7281-9331-1/20/$31.00 ©2020 IEEE

trackers on tracking benchmarks. To enhance the tracking performance, we conduct an analysis of the Siamese network architecture and find out two reasons accounting for this.

Firstly, SiamFC does not make full use of the relationship among images in the video sequence. This tracking method employs a Siamese framework with an exemplar branch and a search branch to merely use the pairwise connection inside samples, while ignores the underlying relationship on the triplet: the first, current and previous frames. Secondly, abundant feature representations from multiple layers are not fully explored in SiamFC. It only captures semantic and abstract information from high-level layers, but does not utilize the outputs of low-level convolutional layers with basic and detailed information whose fine-grained spatial properties also contribute to precise localization. Hence, it is difficult to differentiate target from similar objects and has a high chance to drift when the backgrounds are cluttered.

In this paper, we tackle these issues by adopting the Triplet network as our backbone architecture for efficient and robust tracking. Our Triplet network is made up of three parallel feedforward subnetworks with shared parameters and takes image triplets as input, which takes full advantage of the video sequence and shows its promising performance in tracking applications. Furthermore, we propose to extract multi-level abstraction and combine fine-grained features with semantic features in different layers, which enhances the discriminative capability and allows the tracker to handle the complex background such as similar distractors effectively.

To summarize, the main contributions of this work are illustrated in three-fold.

(1) We define a Triplet network with multi-level feature fusion architecture (TripMFF), which inherits the merits from TripletNet to implement accurate and efficient tracking.

(2) A layer-wise multiple feature fusion structure is designed to fully exploit the shallow, fine features and deep, coarse features of the images, which helps the tracker to distinguish objects from different categories and recognize the complicated background.

(3) The proposed tracker obtains much better performance than many state-of-the-art tracking methods on tracking benchmarks and can operate at real-time speed.

II. Related Work

A. Siamese network-based tracking

Due to the balanced accuracy and speed, Siamese network has drawn great attention in the tracking community recently. By recognizing object tracking as a matching problem, Siamese trackers achieve excellent performance on modern tracking benchmarks. GOTURN [7] applies a regression-based method and leverages the merits of deep networks from offline training to track generic objects at real-time speed. Luca et al. [3] put forward a fully-convolutional Siamese network (SiamFC) to learn a region-wise similarity function from the ILSVRC15 [8] dataset. It contains an exemplar branch and a search branch with shared weight and configuration, which capture the feature representations of the images and generate the response map by the cross-correlation of two output feature maps. CFNet [9] combines the correlation filters (CF) as a differentiable layer with the Siamese architecture trained offline, thereby enabling the CNN-CF combination and resulting in improved performance at high framerates.

B. Combination of multi-level feature representations

In the past few years, it has been proven beneficial to aggregate hierarchical feature maps of different convolutional layers in many computer vision tasks. In the object detection field, Kong et al. [10] develop a deep hierarchical network named HyperNet to concatenate semantic, complementary, and high-resolution features from deep, intermediate, and shallow layers in CNNs for region proposal generation and object detection, which illustrates substantial improvement over other top-ranking detectors. Likewise, a similar network structure is adopted in the semantic segmentation domain. In [11], the authors define a skip architecture that fuses abstract features from a coarse, high layer with appearance features from a fine, low layer to make more abundant feature representations. Regarding to the field of visual tracking, Ma et al. [12] exploit features with semantics and fine-grained details extracted from multi-level of deep CNNs simultaneously for visual tracking. This tracking method significantly mitigates drift and improves tracking accuracy.

C. Triplet network architecture

Triplet network contains three identical CNN branches that share their weights to capture the high-level image semantics and predict the similarity between a pair of images. It was first used in the fine-grained image similarity learning. In [13], the authors introduce a ranking model that calculates the fine-grained image similarity with a set of triplets, and a multi-scale neural network to obtain both the feature representations and global visual characteristics, which performs favorably against other algorithms based on hand-crafted features and deep classification. Hoffer et al. [14] design a Triplet network applying a distance comparison method to extract useful features and conduct a detailed study of the Triplet architecture. Triplet network has many successful applications because of its great performance, so we can exploit its merits in object tracking.

III. PROPOSED METHOD

A. Building block-SiamFC

Tracking an arbitrary object can be considered as a similarity learning problem. SiamFC [3] proposes to use the Siamese network and learn the similarity matching function $f(z, x) = g(\phi(z), \phi(x))$, which compares a template image z to a search image x. If the image z and the image x represent the same object, it will return a high similarity score and conversely a low score. Siamese network works as a feature extractor ϕ to capture the feature representations of the template z and the candidate x, then using predefined metric function g to compare the similarity of representations. The shortcoming of SiamFC is that it only utilizes the semantic information extracted by the high-level convolutional network

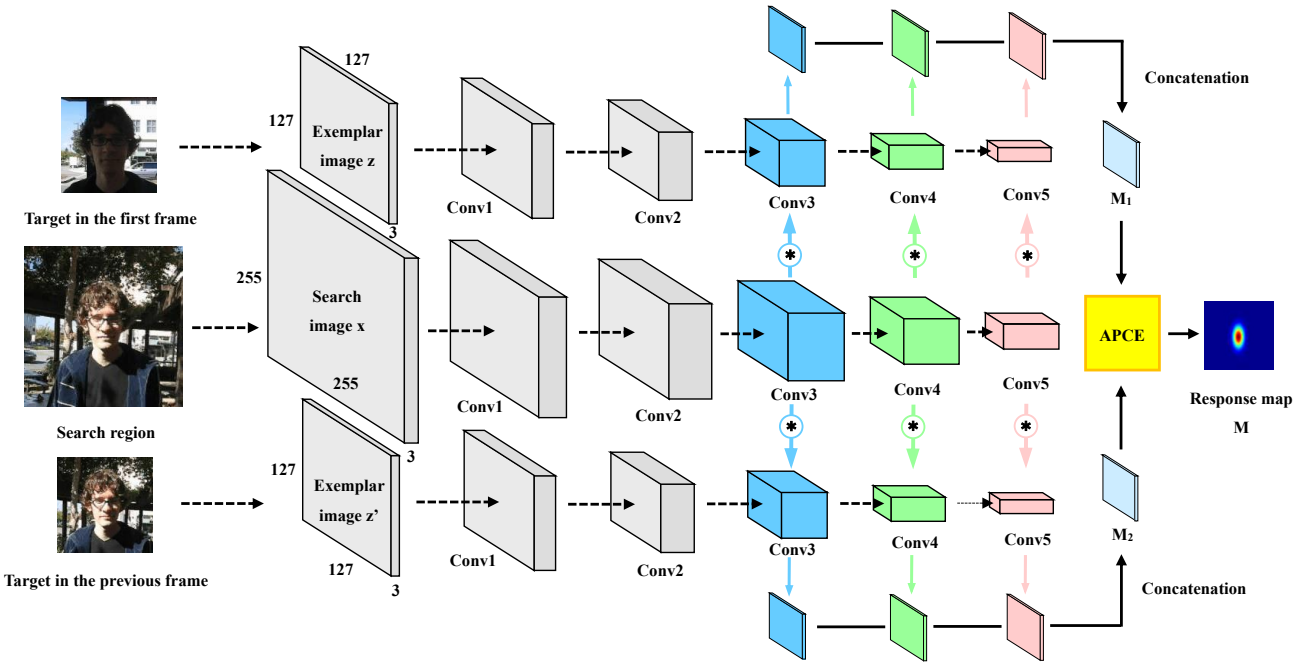


Fig. 2. Overall architecture of TripMFF. It is constituted by three symmetric branches (two exemplar branches and a search branch). When two exemplar images and a search image flow into the network, we capture features from multiple levels and produce the score maps by cross-correlation. APCE measure is used to combine two score maps and obtain the final response map.

and merely chooses the first frame as a template. Therefore, although it can satisfy the real-time requirement and achieve leading performance, the tracker is likely to drift and fails to achieve high performance when similar objects occur in the background.

B. The overall network architecture of TripMFF

In this paper, our tracker uses the Triplet network and takes image triplets as input. One image triplet consists of an exemplar image z from the first frame, an exemplar image z' from the previous frame, and a search image x from the current frame. Our triplet-based network employs three identical fully convolutional neural networks with shared architecture and parameters to extract multi-level feature representations of three input images as shown in Fig. 2, which greatly enriches the hierarchical characteristics of the object features and has higher robustness to track successfully. For clarity, we formulate the measure function as following:

$$f(z, z', x) = h(g(\phi(z), \phi(x)), g(\phi(z'), \phi(x))), \qquad (1)$$

where $\phi(z)$, $\phi(z')$, and $\phi(x)$ denote the score maps which result from the embedding convolution operation of input triplets z, z', and x, function $g(\cdot)$ is the cross-correlation operation and works as the similarity metric of feature representations, function $h(\cdot)$ is the APCE-based fusion strategy which fuses the score maps to generate the final response map.

C. Multi-level feature fusion structure

In order to capture more plentiful and complicated information as well as improve the ability to distinguish similar

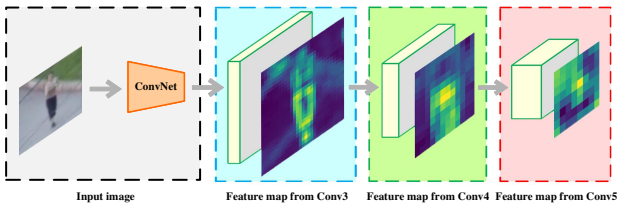


Fig. 3. Visualization of the feature maps from three different convolutional layers. We notice that features extracted by Conv5 are semantic and abstract, while low-level layers like Conv3 and Conv4 encode more spatial and detailed information which is also conducive to accurate localization. Hence, we propose to take advantage of multi-level features for robust object tracking.

distractors, we analyze our Triplet network and visualize the output feature maps of the third, fourth, and fifth layers in Fig. 3. We note that higher layer like Conv5 captures more abstract and semantic features, while lower layers such as Conv3 and Conv4 provide more fine-grained details which also help to locate the target accurately. Therefore, we design a multi-level feature fusion structure and represent the target with the rich feature hierarchies of CNNs.

First of all, we propose to extract features from the last three layers of our Triplet network for similarity metrics. Concretely, Conv3, Conv4, and Conv5 are chosen as our reference layers and three corresponding feature maps are obtained from these convolutional layers. Afterward, we locate an exemplar image within a larger search image to attain score maps with similarity learning. Specifically, the first step is to compute the cross-correlation of feature maps from the same convolutional layer in branch z and branch x to get three scalar-valued score

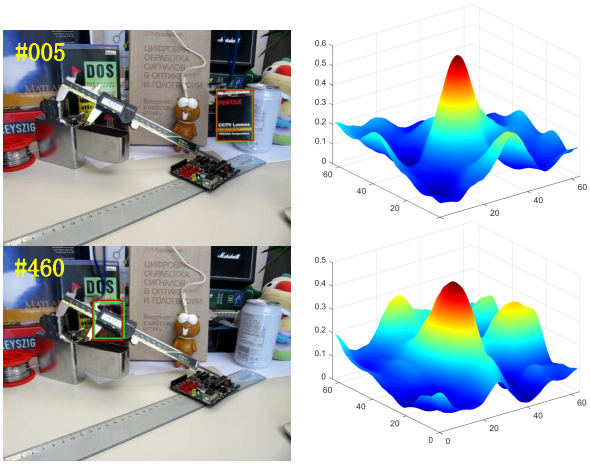


Fig. 4. The first column is the shots of sequence Box from OTB-2015 [15], where the green bounding boxes are the tracking results of TripMFF and the red ones indicate the ground-truth. The response map in the first row of the second column has only one top response which is sharper than the bottom as the object is less occluded. However, the other response map corresponding to the frame with occlusion fluctuates significantly.

maps, and then these maps are concatenated to be the score map M_1, which measures the relative similarity between the first frame and the current frame. Similarly, we compare the similarity of another feature maps from branch z' and branch x as well as acquire a score map M_2. We define score maps M_1, M_2 as follows:

$$M_1 = g(\phi(z), \phi(x)), \quad M_2 = g(\phi(z'), \phi(x)). \tag{2}$$

D. APCE-based fusion strategy

For the purpose of enhancing the localization accuracy and preventing the drift problem caused by distractors or background clutter, we explore a criterion called average peak-to-correlation energy (APCE) measure [16] to reveal the confidence degree of two score maps, which is expressed as:

$$APCE = \frac{|M_{\max} - M_{\min}|^2}{\text{mean}\left(\sum_{w,h}(M_{w,h} - M_{\min})^2\right)}, \tag{3}$$

where $M_{\max}$, $M_{\min}$ and $M_{w,h}$ indicate the maximum, minimum and the w-th row h-th column elements of the response map M. APCE reflects the confidence level of the tracking results. The higher the value is, the more credible the detected target is. Intuitively, as depicted in Fig. 4, a perfect response map has the property of a comparatively high peak at the target position and some small fluctuations in all other areas in the case that the target fully occurs in the search area. In contrast, the response map will fluctuate significantly with several high peaks, and the APCE measure will dramatically decrease when facing occlusion or other challenges such as background clutter and deformation.

After obtaining two score maps M_1 and M_2, the APCE measure of each score map is computed and the final response map M is generated by combining M_1 and M_2. Therefore, the

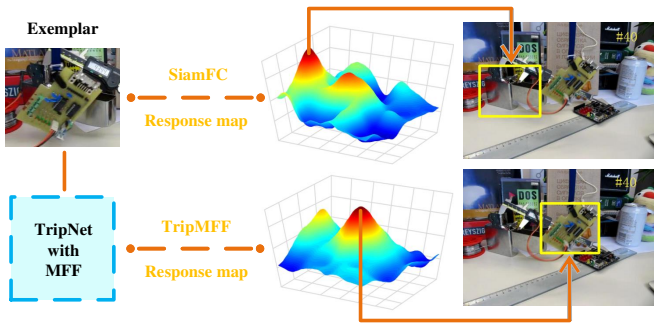


Fig. 5. Illustration of the response maps of SiamFC [3] and TripMFF on the sequence Board from OTB-2015 [15]. As shown in the first row, the response of the target in SiamFC is lower than the background area, which leads to an incorrect tracking result. In contrast, TripMFF generates a more accurate response map where the target stands out and is successfully located.

final fusion response map is formulated as:

$$M = \alpha M_1 + (1-\alpha)M_2, \quad \alpha = \frac{APCE_1}{APCE_1 + APCE_2}, \tag{4}$$

where α and $1-\alpha$ denote the weights of two score maps, $APCE_1$ and $APCE_2$ are the APCE measure of relevant score maps. If the score map fluctuates fiercely with multiple peaks and the corresponding APCE is too small, it represents that the location is not the correct target and the corresponding score map will have a lower weight. Otherwise, it illustrates that the detected target object is completely matched to the correct target object and the score map will get a higher weight. Hence, we can generate a reliable and accurate response map according to the weights of two score maps.

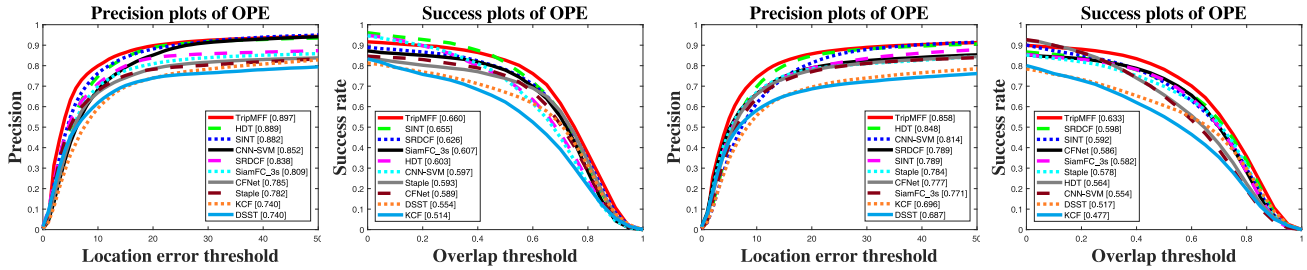
Fig. 5 demonstrates the response maps and tracking results of SiamFC and TripMFF on the sequence Board. In such a complex tracking scenario, SiamFC is disturbed by the cluttered background, thus producing an inaccurate response map and failing to track the correct target. On the contrary, the response of the target in TripMFF is more precise, which enables it to track successfully.

IV. Network Training and Tracking

During the training process, ILSVRC2015 [8] dataset with around 4,000 videos annotated frame-by-frame is utilized to train a network without overfitting. The exact dimensions of feature maps and network parameters are shown in Table II. We pick a template patch z and a candidate patch x as image pairs centered on the target object. They are extracted from two frames with a random interval in the same video and are cropped to 127 $\times$ 127 and 255 $\times$ 255, respectively. We use positive and negative image pairs to train the model and adopt the logistic loss as follows:

$$l(j,i) = \log\left(1 + e^{-ji}\right), \tag{5}$$

where i represents the real-valued score of a single template-search pair, and j means its ground-truth label. This function denotes the possibility of the search patch matched to the target object, which makes positive pairs closer and negative pairs farther. The smaller the loss is, the higher the confidence of the



(a) Comparisons on OTB-2013 (b) Comparisons on OTB-2015

Fig. 6. Comparison on OTB-2013 and OTB-2015 using distance precision rate (DPR) and overlap success rate (OSR).

TABLE I

COMPARISONS WITH STATE-OF-THE-ART TRACKERS ON OTB-2013 [17] AND OTB-2015 [15]. OUR TRIPMFF PERFORMS FAVORABLY AGAINST EXISTING METHODS IN DISTANCE PRECISION RATE (DPR) AT A THRESHOLD OF 20 PIXELS AND OVERLAP SUCCESS RATE (OSR) AT AN OVERLAP THRESHOLD OF 0.5.

		Deep learning trackers					Correlation filters trackers				Combination tracker
		TripMFF (ours)	HDT [18]	SINT [19]	SiamFC [3]	CNN_SVM [20]	SRDCF [21]	Staple [22]	DSST [23]	KCF [24]	CFNet [9]
OTB2015	DPR (%)	89.7	88.9	88.2	80.9	85.2	83.8	78.2	74.0	74.0	78.5
	OSR (%)	82.7	73.7	81.6	77.9	73.4	78.1	73.8	67.0	62.3	75.2
OTB2015	DPR (%)	85.8	84.8	78.9	77.1	81.4	78.9	78.4	68.7	69.6	77.7
	OSR (%)	78.2	65.7	71.9	73.0	65.1	72.8	69.9	60.5	55.1	73.7

patch to the target is. For each image pair fed to the network, a response map D was obtained. We define the loss function of the response map to be the mean of the logistic loss of individual element between the labeled value $j[u]$ and ground-truth value $i[u]$, which is depicted as:

$$L(j,i) = \frac{1}{|D|}\sum_{u\in D} l(j[u], i[u]). \tag{6}$$

We apply the SGD (Stochastic Gradient Descent) algorithm to minimize the loss function illustrated in Eq. (7) and get the network parameters.

$$\arg\min_{\theta} \underset{(z,x,j)}{E} L(j, f(z,x;\theta)). \tag{7}$$

The parameters of the network stay constant when online tracking. For each new image, the exemplar branches are used to compute features for the target from the first frame and previous frame in the video, and the search branch computes feature for search region and cross-correlation with features of exemplar branches. The two score maps are fused by APCE to generate the final response map and the new target position is identified by the location of maximum on the response map.

V. EXPERIMENT

We conduct the performance evaluation of the TripMFF tracker and some state-of-the-art tracking approaches on the benchmark datasets. Our tracking algorithm is implemented in PyTorch 0.4.1 framework on a machine with a NVIDIA GeForce GTX 2080 GPU and an Intel Xeon E5 2.5GHz CPU. The average testing speed of TripMFF is around 30 fps.

A. Implementation detail

TripMFF is trained offline over 50 epochs applying stochastic gradient descent (SGD). Most of the parameters in the training and tracking phase are set the same as SiamFC [3]. We use Kaiming Normal Initialization [25] to initialize the parameters of our network. The momentum is 0.9, the weight decay is 0.0005, and the learning rate is set from 10^{-2} to 10^{-5}. In order to handle scale transformation, we exploit three scales: $1.025^{\{-1,0,1\}}$ to search for the object. We also upsample the response map from 17×17 to 272×272 using bicubic interpolation for more accurate localization.

B. Experiments on OTB-2013 and OTB-2015

1) Evaluation on OTB dataset: The experiments are conducted on OTB-2013 [17] and OTB-2015 [15] benchmarks, which comprise a number of fully annotated videos with various attributes. We evaluate the TripMFF on these benchmarks with comparisons to 9 excellent trackers. These tracking methods can be approximately classified into three categories: (i) deep learning trackers, including CNN-SVM [20], HDT [18], SiamFC [3] and SINT [19]; (ii) correlation filters trackers, including SRDCF [21], DSST [23], KCF [24] and Staple [22]; (iii) tracker combining both deep features and correlation filters: CFNet [9]. It is worth mentioning that our tracking algorithm TripMFF is on the basis of SiamFC, so it can be regarded as our baseline.

Following [15], [17], we employ the one-pass evaluation (OPE) to compare the tracking results of different trackers, and two metrics: distance precision rate (DPR) and overlap success rate (OSR) are exploited to conduct quantitative and qualitative evaluations. Experimental results for these tracking approaches

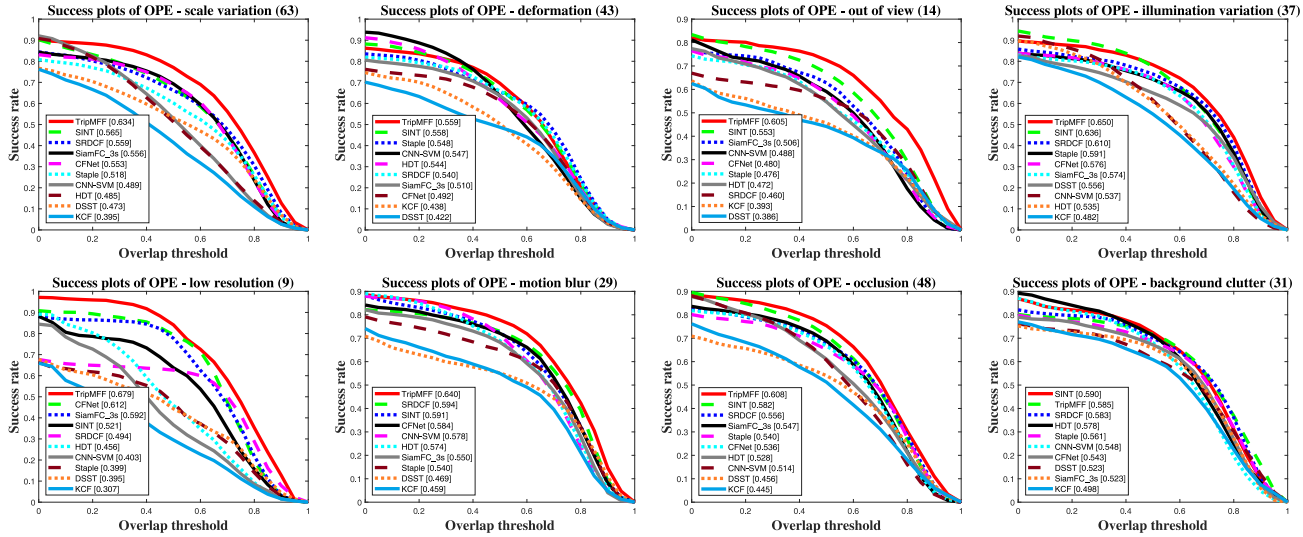


Fig. 7. The success plots over eight tracking challenges, including scale variation, deformation, out-of-view, illumination variation, low resolution, motion blur, occlusion, and background clutter.

TABLE II
THE BACKBONE ARCHITECTURE OF TRIPMFF. EFMS STANDS FOR THE SIZE OF FEATURE MAP FOR EXEMPLAR IMAGE, WHILE SFMS FOR SEARCH IMAGE. THE CHANNEL MAP MEANS THE NUMBER OF OUTPUT AND INPUT CHANNELS USING THE FORMAT $output\ channel \times input\ channel$.

Layer	Kernel size	Chan.map	Stride	EFMS	SFMS	Chans.
				127 × 127	255 × 255	×3
Conv1	11 × 11	96 ×3	2	59 × 59	123 × 123	×96
Pool1	3 × 3		2	29 × 29	61 × 61	×96
Conv2	5 × 5	256 × 48	1	25 × 25	57 × 57	×256
Pool2	3 × 3		2	12 × 12	28 × 28	×256
Conv3	3 × 3	384 × 256	1	10 × 10	26 × 26	×192
Conv4	3 × 3	384 × 192	1	8 × 8	24 × 24	×192
Conv5	3 × 3	256 × 192	1	6 × 6	22 × 22	×128

are illustrated in Fig. 6. In general, the proposed tracker outperforms all other top-ranking methods on both benchmarks. Furthermore, a quantitative comparison of DPR at 20 pixels and OVR at 0.5 is presented in Table I. It demonstrates the superiority of our algorithm among these tracking methods. In specific, we obtain a DPR of 85.8% and an OVR of 78.2% on OTB-2015. Compared with HDT, although it applies multi-layer features and develops an improved Hedge algorithm for visual tracking, our method obtains 1.0% gains on DPR and 12.5% gains on OVR. Besides, our TripMFF tracker achieves a much faster tracking speed than HDT (10 fps). Our tracker also performs better against SINT by improving 6.9% on DPR and 6.3% on OVR. In addition, TripMFF runs in real-time while SINT operates at a very slow speed for around 4 fps. In comparison with the baseline SiamFC with 77.1% DPR and 73.0% OVR, our TripMFF enhances 8.7% on DPR and 5.2% on OVR, showing the advantages of the multi-level feature fusion and Triplet network architecture in accurate localization.

2) Attribute-based Comparison: To analyze the performance of our method in a variety of scenarios, we evaluate TripMFF and the top-ranking trackers under different attributes on OTB-2015 [15]. As demonstrated in Fig. 7, our tracker produces a leading result in most tracking attributes. In particular, TripMFF gets a better performance than SiamFC in all the challenging factors such as motion blur, scale variation, and low resolution, which indicates that our approach is more robust than its foundation tracker. The reasons that the proposed method achieves promising results can be explained by two aspects. On one hand, we employ the Triplet network as our backbone architecture, which makes the exemplar branches have more sufficient and discriminative information. One the other hand, multi-level CNN features including fine-grained details from earlier layers and category-level semantics from later layers are fused to assist in tracking accuracy. In contrast, SiamFC only utilizes the pairwise connection in image sequences and merely takes advantage of the last convolutional layer. Therefore, its performance is suffered in complex tracking scenarios.

Fig. 8. Qualitative evaluation of our TripMFF tracker, SiamFC [3], HDT [18], Staple [22], KCF [24], and DSST [23] in twelve difficult sequences (from left to right and top to down: Basketball, Football, Bird1, Shaking, Lemming, Coke, Jogging-1, Jumping, Panda, Matrix, Skiing, and MotorRolling, respectively).

3) Qualitative Analysis: We further provide some tracking results of the top-ranking tracking algorithms: KCF [24], DSST [23], Staple [22], HDT [18], SiamFC [3], and the proposed method in 12 difficult sequences in Fig. 8. The correlation filters trackers (Staple, KCF, and DSST) track the target well in the presence of illumination variation (Shaking) and partial occlusion (Basketball), but they are prone to drift in sequences with rotation (MotorRolling and Skiing). As for HDT, it does not look for the target in a whole frame, and thus HDT may lose the target in sequences with out-of-view (Bird1). Although SiamFC adopts a similarity learning method and performs well on scale variation (Lemming), motion blur (Jumping), and occlusion (Jogging-1), it generally falls in failure when similar objects appear, such as Matrix, Basketball, and Football. This mainly attributes to that its extracted features are not discriminative enough to differentiate the target and distractors. Our tracker proposes to utilize the Triplet network and combine abstract semantics and spatial details to create more abundant features. As a result, our algorithm can predict a more precise response map for localization and performs well in all the twelve sequences.

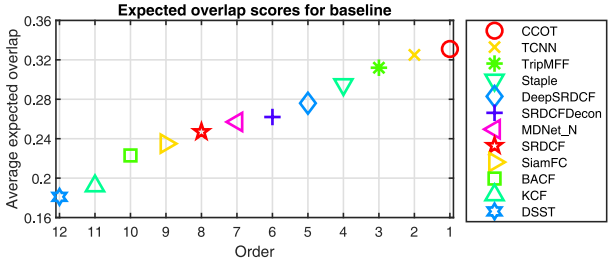


Fig. 9. Comparisons on VOT-2016 [26] using EAO. Larger value demonstrates better performance. Our TripMFF performs favorably against the baseline and ranks in the 3rd place.

C. Experiments on VOT-2016 and VOT-2017

1) Evaluation on VOT-2016 dataset: VOT-2016 dataset [26] consists of 60 sequences with 5 different challenges. It incorporates the re-initialization settings and concentrates much more on the short-term visual tracking. Expected Average Overlap (EAO) considering both robustness (failure rate) and accuracy (average overlap in successful tracking) is the major metric to evaluate the overall performance of a tracking

TABLE III
DETAILED COMPARISONS ON VOT-2016 [26]. **RED**, **GREEN**, AND **BLUE** FONTS DENOTE THE TOP-3 TRACKERS, RESPECTIVELY.

Tracker	EAO	Accuracy	Robustness	FPS
TripMFF	**0.312**	**0.56**	**0.93**	30
CCOT	**0.331**	0.54	**0.85**	0.3
TCNN	**0.325**	**0.55**	**0.96**	1.5
Staple	0.295	0.54	1.35	**80**
DeepSRDCF	0.276	0.51	1.17	0.2
SRDCFDecon	0.262	0.53	1.42	2.3
MDNet_N	0.257	0.53	1.20	1
SRDCF	0.247	0.52	1.50	5
SiamFC	0.235	0.53	1.91	**86**
BACF	0.223	**0.56**	1.88	35
KCF	0.192	0.48	1.95	**172**
DSST	0.181	0.50	2.72	24

TABLE IV
COMPARISONS ON VOT-2017 [32] USING EAO. **RED**, **GREEN**, AND **BLUE** FONTS DENOTE THE TOP-3 TRACKERS, RESPECTIVELY.

Tracker	Baseline EAO	Real-time EAO
TripMFF	**0.221**	**0.218**
DLST	**0.233**	0.057
UCT	0.206	0.145
MEEM	0.192	0.072
SiamFC	0.188	**0.182**
Staple	0.169	**0.170**
ASMS	0.169	0.168
ANT	0.168	0.059
DPT	0.158	0.126
LGT	0.144	0.059
KCF	0.135	0.134
SRDCF	0.119	0.058

algorithm.

We compare our TripMFF tracker with 11 state-of-art tracking methods, including CCOT [27], TCNN [28], Staple [22], SRDCF [21], DeepSRDCF [29], SRDCFDecon [30], MDnet_N [4], BACF [31], KCF [24], DSST [23], and the baseline SiamFC [3] on VOT-2016 [26]. Fig. 9 illustrates the EAO of different trackers. TripMFF achieves competitive results and performs favorably against its baseline SiamFC. Table III lists the performance of different trackers on VOT2016. It is worth noticing that TripMFF ranks 3rd in EAO, 1st in accuracy, and 2nd in robustness while operating at high speed.

2) Evaluation on VOT-2017 dataset: VOT-2017 [32] is developed from VOT-2016 [26], which updates the sequences by replacing 10 easy videos with 10 challenging sequences. In addition, a new real-time experiment is introduced to analyze both tracking performance and efficiency of different trackers, which means that the tracker needs to achieve high tracking accuracy while satisfying the requirement of real-time.

We compare our TripMFF tracker with 11 top-ranking tracking approaches, including DLST [33], UCT [34], MEEM [35], Staple [22], ASMS [36], ANT [37], DPT [38], LGT [39], KCF [24], SRDCF [21], and the baseline SiamFC [3] on VOT-2017 [32] applying the EAO of baseline and real-time experiments. Table IV shows TripMFF obtains the EAO score of 0.218, performing remarkably better than SiamFC with 0.188. Besides, TripMFF has the best performance in terms of real-time EAO, which demonstrates advances in accuracy, robustness, and speed.

D. Ablation Studies

We implement the ablation studies to verify the effect of each element in our tracker on OTB-2015 [15], and present the comparison results in Fig. 10. The basic concepts are as follows. (1) 'Baseline' means the SiamFC tracker that exploits the fully-convolutional Siamese network. (2) 'Baseline+Trip' denotes the baseline algorithm adopting the Triplet network with an APCE-based fusion strategy. (3) 'Baseline+MFF' stands for the baseline approach with adding a multi-level feature fusion structure. (4) 'Baseline+Trip+MFF' is our proposed method that combines the baseline algorithm with both multi-level feature fusion structure and Triplet architecture.

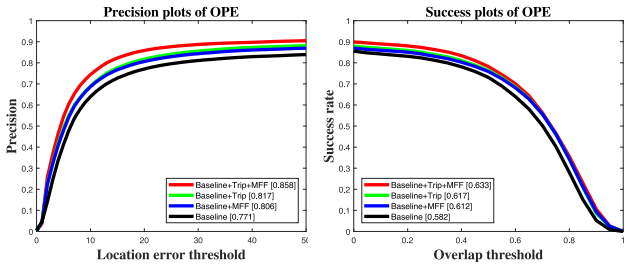


Fig. 10. Experimental results for each element of the proposed tracker.

As shown in Fig. 10, we can see that both the Triplet network with an APCE-based fusion strategy and multi-level feature fusion structure contribute to the considerable improvement over the baseline tracker. In addition, our proposed method improves the baseline tracker by relative enhancements of 8.7% in precision plots as well as 5.1% in success plots.

VI. CONCLUSION

In this paper, we put forward a Triplet network with multi-level feature fusion (TripMFF) which is trained offline with image pairs from ILSVRC. Particularly, the Triplet network with triplet inputs and three parallel feedforward subnetworks is adopted as our backbone architecture to make full use of the relationship among images in the video sequence. Meanwhile, we develop a multi-level feature fusion structure to integrate semantic and abstract information from high-level layers with fine-grained and spatial information from low-level layers, which improves the discriminative capability and enables TripMFF to achieve more robust performance in distinguishing the true target from the misleading distractors. Performance evaluation on popular tracking benchmarks indicate that our tracking method TripMFF outperforms other top-ranking trackers in accuracy and efficiency.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[2] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[4] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4293–4302.

[5] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Stct: Sequentially training convolutional networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1373–1381.

[6] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2711–2720.

[7] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 749–765.

[8] 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.

[9] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2805–2813.

[10] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 845–853.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[12] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3074–3082.

[13] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[14] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[15] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[16] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4021–4029.

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

[18] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4303–4311.

[19] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1420–1429.

[20] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International conference on machine learning*, 2015, pp. 597–606.

[21] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4310–4318.

[22] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1401–1409.

[23] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[26] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, and T. Vojir, "Hager, and et al. the visual object tracking vot2016 challenge results," in *ECCV workshop*, vol. 2, no. 6, 2016, p. 8.

[27] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European conference on computer vision*. Springer, 2016, pp. 472–488.

[28] H. Nam, M. Baek, and B. Han, "Modeling and propagating cnns in a tree structure for visual tracking," *arXiv preprint arXiv:1608.07242*, 2016.

[29] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 58–66.

[30] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1430–1438.

[31] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1135–1143.

[32] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey *et al.*, "The visual object tracking vot2017 challenge results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1949–1972.

[33] L. Yang, R. Liu, D. Zhang, and L. Zhang, "Deep location-specific tracking," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1309–1317.

[34] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang, "Uct: Learning unified convolutional networks for real-time visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1973–1982.

[35] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *European conference on computer vision*. Springer, 2014, pp. 188–203.

[36] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250–258, 2014.

[37] L. Čehovin, A. Leonardis, and M. Kristan, "Robust visual tracking using template anchors," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–8.

[38] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE transactions on cybernetics*, vol. 48, no. 6, pp. 1849–1861, 2017.

[39] L. Cehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 941–953, 2012.