Object Tracking using Spatio-Temporal Networks for Future Prediction Location

Yuan Liu[1] [0000−0001−7514−8876]⋆, Ruoteng Li[2] [0000−0001−5962−9477]⋆, Yu Cheng[2] [0000−0002−9830−0081], Robby T. Tan[2,3] [0000−−0001−7532−6919]⋆⋆, and Xiubao Sui[1] [0000−0003−0464−464X]

[1] Nanjing University of Science and Technology
[2] National University of Singapore
[3] Yale-NUS College
walkyuan90@gmail.com, {liruoteng, e0321276}@u.nus.edu,
robby.tan@nus.edu.sg, sxbhandsome@njust.edu.cn

Abstract. We introduce an object tracking algorithm that predicts the future locations of the target object and assists the tracker to handle object occlusion. Given a few frames of an object that are extracted from a complete input sequence, we aim to predict the object's location in the future frames. To facilitate the future prediction ability, we follow three key observations: 1) object motion trajectory is affected significantly by camera motion; 2) the past trajectory of an object can act as a salient cue to estimate the object motion in the spatial domain; 3) previous frames contain the surroundings and appearance of the target object, which is useful for predicting the target object's future locations. We incorporate these three observations into our method that employs a multi-stream convolutional-LSTM network. By combining the heatmap scores from our tracker (that utilises appearance inference) and the locations of the target object from our trajectory inference, we predict the final target's location in each frame. Comprehensive evaluations show that our method sets new state-of-the-art performance on a few commonly used tracking benchmarks.

Keywords: Object tracking, trajectory prediction, background motion

1 Introduction

Object tracking is important for many computer vision applications, such as surveillance, vehicle navigation, privacy preservation, activity recognition, etc.

⋆ These two authors contributed equally to this work.

⋆⋆ Robby T. Tan's research in this work is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

While significant progress has been made recently, object tracking is still challenging due to a few factors such as: illumination variation, occlusion, background clutters and so on [30]. Given a target object indicated at first frame of an input video, the aim of visual object tracking is to estimate its positions in all the subsequent frames [28, 35, 32]. Recently, a Siamese network based trackers[3, 29, 17, 37] have drawn attention in the field. The Siamese trackers cast the visual object tracking problem as learning a general similarity function by computing cross-correlation between the feature representations learned for the target template and the search region. Based on the efficiency of the Siamese network and the feature representations of the convolutional network, the Siamese trackers obtain good tracking performance.

Despite the progress, however, most of existing methods including Siamese trackers tend to fail in tracking an occluded target object and are erroneous for multiple objects with similar appearance [14, 30]. We observe that most trackers focus on improving target object's feature representation by deep convolutional networks. A good feature representation is important, however it can be problematic when target is occluded or when there are similar-looking objects nearby. Instead of making full use of the target object observation in the previous few frames, many of these methods utilize the target's location in the previous immediate frame to reduce the search region or to update the representation model [34].

To address the problem, we aim at leveraging the predicted future trajectory or future locations to deal with occlusion. When the target suffers from severe occlusion, there is little useful information in the spatial domain at the current frame to detect the target object. Hence, our basic idea is that, when the target object is severely occluded, the predicted future trajectory should be critical information to correct tracker's estimation. In other words, when the tracker lose the target object due to occlusion, our predicted future location based on the target's past trajectory is more proper information than the prediction of the low-confident tracker. Based on this idea, we develop a trajectory-guided deep network that predicts the target's possible locations in future frames.

To realise the idea, we consider the following three key observations. First, camera motion significantly affects the background motion, and thus the target object's locations in the image frames. This camera motion should be incorporated into the future trajectory prediction. Second, the target object's past trajectory can act as a salient cue to estimate the target object's motion in the spatial domain. Third, previous frames contain the surroundings and appearance of the target object, which is useful for predicting the target object's future locations.

Based on these key observations, we propose a method to predict the target obect's future locations based on the camera motion, the location of the target object, and the past few frames. Our method consists of 3 networks: an appearance-based tracking network (tracker), a background-motion prediction network, and a trajectory prediction network. The tracker provides the estimated target object's locations from appearance inference, which is useful particularly when occlusion does not occur. The background-motion prediction network captures the camera motion to compensate the target object's trajectory in the input video. The trajectory prediction network predicts the target object's future locations from the target's past observations. When occlusion happens, the trajectory-guided tracking mechanism is used to avoid drifting, making our approach switch dynamically between the tracker and the trajectory prediction states. To summarise, in this paper, our main contributions are as follows:

- We introduce a background motion model that captures the global background motion between adjacent frames to represent the effect of camera motion on image coordinates. This background motion is important to compensate the motion of the camera.
- We propose a new trajectory prediction model that learns from the target object's observations in several previous frames and predicts the locations of the target object in the subsequent future frames. A multi-stream conv-LSTM architecture is introduced to encode and decode temporal evolution in these observations.
- We present a trajectory-guided tracking mechanism by using a trajectory selection score, which helps the tracker to switch dynamically between the current tracking status and our trajectory predictor, particularly when the target object is occluded.

2 Related Works

Visual Object Tracking A tracking-by-detection paradigm [2] is introduced to train a discriminative classifier from the ground-truth information provided in the first frame and update it online. By comparing the template of an arbitrary target and its 2D translations, the correlation filter [6] is employed for its speed and effective strategy for tracking-by-detection. A number of methods based on the correlation filter improve the tracking performance with the adoption of multi-channel formulations [17], spatial constraints [10] and deep features [9]. Recently, a few methods use a fully-convolutional Siamese approach [3, 29, 17, 37]. Instead of learning a discriminative classifier online, the approach aims to learn a similarity function offline on pairs of video frames. Then, this similarity function is simply evaluated online once per frame during the tracking process. On the basis of this, a number of methods improved tracking performance by making use of region proposals [24], hard negative mining [37] and binary segmentation [29]. Some methods employ temporal information for better object feature representation. Yang et al. in [33] feed the target object's image patch to a Recurrent Neural Networks (RNN) to estimate an object-specific filter for tracking. Cui et al. in [7] propose a multi-directional RNN to capture long-range contextual cues by traversing a candidate spatial region.

Most modern trackers focus on modelling the object feature representation to track a single target in different frames. It proves that the feature representation is an important and effective way to improve tracking performance. However, relying only on object feature representation can be problematic in cases where the target is occluded or the target come across other objects with similar appearance. To this end, we combine the object representation in the spatial domain with the trajectory prediction in the temporal domain by using a spatio-temporal network to track target accurately and robustly.

Trajectory Prediction Unlike the object tracking problem, future trajectory prediction focuses on predicting target's positions in future frames [1]. Recently, a large body of works focus on person trajectory prediction by considering human social interactions and behaviors in crowded scene. Zou et al. in [38] learn human behaviors in crowds by using a decision-making process. Liang et al. in [18] utilize rich visual features about human behavioral information and interaction with their surroundings to predict future path jointly with future activities. A number of methods try to learn the effects of the physical scene. Scene-LSTM [21] divides the static scene into Manhattan grid and predict pedestrian's location using LSTM. SoPhie [27] combines deep-net features from a scene semantic segmentation model and generative adversarial network using attention to model person trajectories.

There are some methods that take the motion prediction into account for tracking or predicting person path. Amir et al. in [26] propose a structure of RNN that jointly reasons on multiple cues over a temporal window for multi-target tracking. Ellis et al. in [12] propose a Gaussian process regression model for pedestrian motion. Hogg et al. in [13] propose a statistically based model of object trajectories which is learned from image sequences. Compared with these methods, which assume a static camera in modeling the trajectory, our idea is to integrate trajectory prediction into object tracking problem using deep learning for a dynamic camera. To simplify the trajectory complexity, several methods consider motion as camera motion and object motion. In particular, Takuma et al. in [31] recently proposed an accurate method that makes use of camera ego-motion, scales and speed of the target person, and person pose to predict person's location in future frames. Unlike these methods that use object surroundings, which are expensive for general single object tracking, we utilize only the past trajectory and target visual features to predict short-term future locations to assist the tracker.

3 Proposed Method

As shown in Figure 1, our approach consists of 3 networks: an appearance-based tracking network (tracker), a background-motion prediction network, and a trajectory prediction network. Given frame t, the tracker estimates the target's location l_{track} based on appearance inference. To compensate the target object's trajectory to the current camera coordinate system, the background-motion prediction network captures the global background motion vector $v_{in}^{'}$ between previous adjacent frames. Based on the background motion, the target object's previous locations, and a few previous images, our trajectory prediction network predicts the target's future location l_{traj} and also outputs the confidence

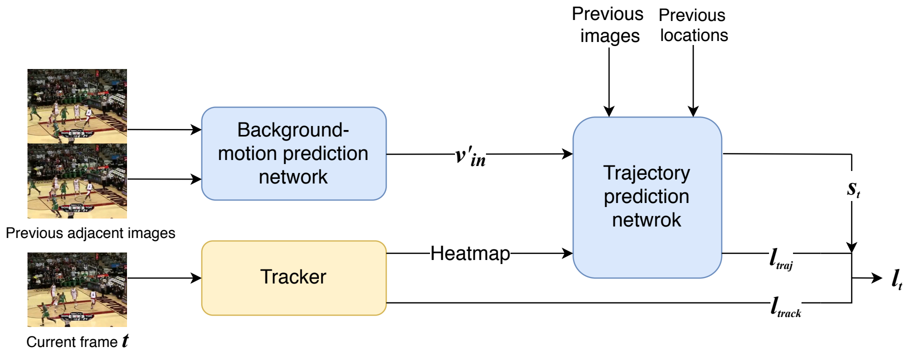

Fig. 1. The whole architecture of our approach. Our approach consists of 3 networks: tracker, a background-motion prediction network, and a trajectory prediction network.

score s_t of this prediction. The final estimated location l_t is selected depending on s_t from l_{traj} and l_{track}.

3.1 Tracker Module

Let $l_t \in \mathbb{R}^2_+$ be the 2D location of the target at the frame t . Given the location l_0 of an arbitrary target labeled at the first frame of a video, the tracker's task is to estimate its position in the current frame, t. By comparing an exemplar image patch with a larger search region image, the tracker [3] produces a heatmap, g, from which the the estimated position at the current frame l_t can be obtained based on the maximum value. We compute $g = f(x) * f(z)$, where, z and x are, respectively, a crop centered on the target object and a larger crop centered on the last estimated position l_{t-1} of the target. The operator $*$ denotes the cross-correlation and f represents the convolutional network mapping of the tracker module. While this tracker can work properly, unfortunately it tend to fail when the target object is severely occluded in a number of consecutive frames. Since the target object's appearance is totally hidden by the severe occlusion, this tracker is unable to obtain any useful visual information from the current image.

To address this occlusion problem, our basic idea is that the occluded target object's location can be more reliably predicted using its past trajectory information, since the current frame is unreliable. As illustrated in Figure 2, we aim at predicting the target's locations in the current frame t and subsequent t_{future} frames (the red boxes the Figure 2), namely: $l_{out} = (l_t, ..., l_{t+t_{future}})$, based on observations $l_{in} = (l_{t-t_{prev}}, ..., l_{t-1})$ in the previous t_{prev} frames (the blue boxes). When the target object is severely occluded in a number of consecutive frames, the estimated target object's location will follow our prediction trajectory l_{out} from frame t to the future frame t_{future}.

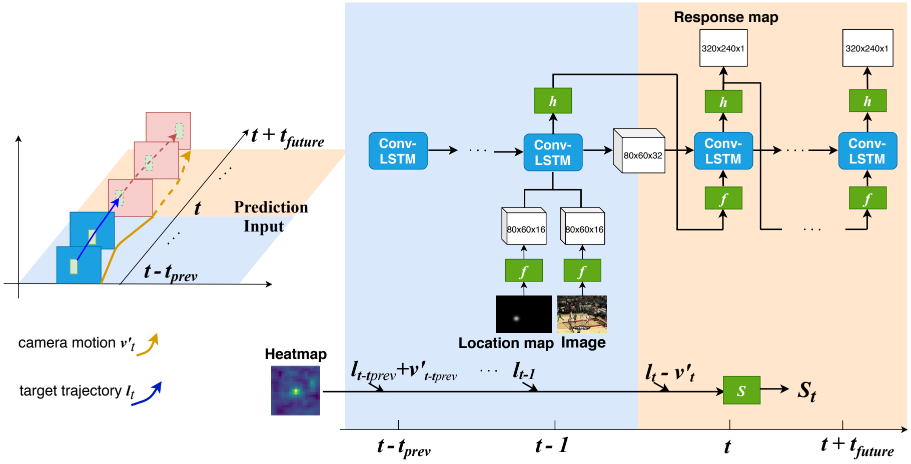

Fig. 2. The architecture of trajectory prediction. Given t_{prev} frames observations as input, we predict future locations of a target in the current t frame and subsequent t_{future} frames. f denote convolutional operation, h denote deconvolutional operation, s denote 1-D convolutional operation

3.2 Background Motion

While the target object's past locations explicitly show how the object is likely to move over time, predicting l_{out} directly from l_{in} is problematic due to significant camera motion present in the input video. More specifically, the coordinate system to describe each point l_t changes dynamically as the camera moves. This causes the correlation between l_{out} and l_{in} to be complex, as it depends on both the trajectory of the target object and camera motion.

To improve future localization performance, we need to estimate camera motion's parameters. Specifically, the camera motion between adjacent frames, which can be represented by rotation and translation. Rotation is described by a rotation matrix $R_t \in \mathbb{R}^{3\times3}$ and translation is described by a vector $V_t \in \mathbb{R}^3$,(i.e., x-, y-, z-axes), both from frame $t-1$ to frame t in the camera coordinate system at frame $t-1$.

However, the accurate acquisition of these vectors is difficult without the image depth information. Our solution is to obtained the approximated camera motion from the adjacent frame directly. Intuitively, camera motion is observed in the form of global background motion of object tracking videos. Detecting the camera motion can be simplified as detecting the global background motion between the adjacent frames. We simplify the rotation matrix R_t to one rotation angle r_t in the image domain, translation vector V_t to the translation vector $v_t \in \mathbb{R}^2$,(i.e., x-, y-axes) and scale changing c_t.

For detecting the global background motion, we employ a Siamese network that compares the adjacent frames as shown in Figure 3. Different from the

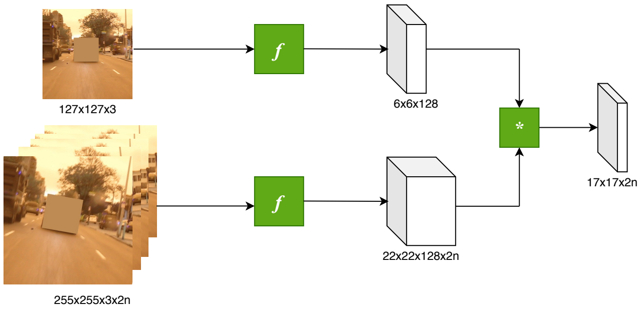

Fig. 3. The architecture of background motion model. We utilize the Siamese network to compare the background between the adjacent frames. f denotes the convolutional network embeding, and $*$ denotes the cross-correlation. To estimate the scale and rotation changing, we search $2n$ image patches by using patch pyramid with different scale and rotation factors.

Siamese network based tracker, we focus on comparing the similarity of the global background instead of the target object. Thus, the exemplar image z_{t-1} is cropped at the center of input image in the frame $t-1$, and the search image x_t is the larger cropped image patch in the frame t. To avoid the interference of the target object's motion, the target region is masked as one value (i.e., the average value of the whole image). The heatmap g_t of matching background in the frame t can be achieved by:

$$g_t(z_{t-1}, x_t) = f(x_t) * f(z_{t-1}). \tag{1}$$

To estimate the scale and rotation changing, we also search image patches $x_{c1}, ..., x_{cn}$ and $x_{r1}, ..., x_{rn}$, by using patch pyramid with different scale and rotation factors. Finally, the displacement of the maximum value position relative to the center in the heatmap is the background motion translation vector v_t. The position of the maximum value in the multi-scales heatmap set denotes the scale changing c_t, and the position of the maximum value in the multi-rotations heatmap set denotes the rotation changing r_t.

These vectors that represent the local movement between two adjacent frames, and do not capture the global movement along multiple frames. Therefore, for each frame within the input interval $[t_0 - t_{prev}, t_0 - 1]$, we accumulate those vectors to describe the time-varying background motion patterns in the global background coordinate system at frame $t_0 - 1$:

$$v'_t = \begin{cases} r_t c_t v_t & (t = t_0 - 2) \\ r_t c_t v_t + v'_{t+1} & (t < t_0 - 2) \end{cases} \tag{2}$$

where v'_t denotes the background motion patterns of frame t in the global background coordinate system at frame $t_0 - 1$. $t \in [t_0 - t_{prev}, t_0 - 1]$. v_t, r_t and c_t denotes translation vector, rotation and scale changing between two adjacent frames t and $t-1$.

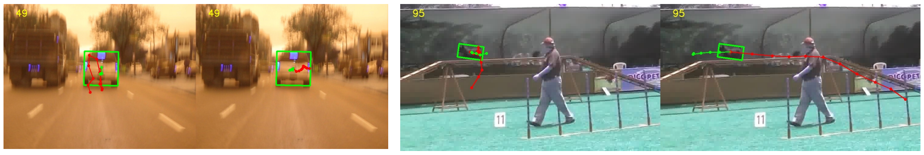

Fig. 4. Examples of the target object's original trajectory and the trajectory that compensates background motion v'_t. In each example, the left image is the target's original trajectory and the right one is the one with background motion correction. The red line is the target object's past trajectory and the green line is the target object's future trajectory.

3.3 Trajectory Prediction

Most of existing object tracking methods tend to fail when the target object is severely occluded . It is challenging to extract useful location information from the current video frame when the target does not exist visually. We have to rely on the target's past observations to predict future possible locations.

Intuitively, the straightforward way to predict future locations of the target object is to utilize its previous immediate location. However, it is insufficient for the object tracking problem, because the target object's motion can be arbitrary and affected by camera motion. To solve this, we separate the motion of the target object from camera motion, where the latter can be captured as the global background motion vector, mentioned in Section 3.2. In order to show the target motion in the spatial domain, we utilize a 2D location map to represent the target object's location in each frame. The location map is a Gaussian function, where the peak locates the target object and other positions locates the background. Therefore, the location map is able to provide the target object's location and also can be seen as a response map that provides high values in the target object's region.

Based on the discussions we made in Section 1 (also refer to Figure 2), we focus on the location map of the target object, background motion, and input image frame as the cues to approach the problem. To predict future locations from those cues, we develop a fully-convolutional network that utilizes a multi-stream conv-LSTM architecture shown in Figure 2. The location map and input image frame are extracted by a two-stream convolutional network f with different learnable parameters μ, θ respectively. Given a sequence of the concatenated features provided from all input streams, some response maps are deconvoluted after the encoding and decoding of LSTM. The overall network can be trained end-to-end via back-propagation.

Let T denotes location map and I denotes image. The predicted response map p can be obtained by:

$$p = h_\sigma(m_\psi(f_\mu(T), f_\theta(I))), \tag{3}$$

where, h_σ denotes the deconvolutional tranfermation with parameter σ, m_ψ simply indicates conv-LSTM with parameter ψ, f_μ and f_θ representing the two stream convolutional networks embeding with parameter μ and θ.

The predicted response map is labelled with a Gaussian function peaked at the target object's location. Let us denote the label y_t and the predicted response map p_t in the frame t . The loss function L_{pred} for the trajectory prediction task is a L1 loss over all predicting future frames:

$$L_{pred} = \sum\nolimits_{t}^{t_{future}} ||p_t - y_t||. \tag{4}$$

Trajectory Selection Since there are the trajectory prediction result and the tracking result at one frame, a selection mechanism is needed to compare a more correct location between tracking location and prediction location. This can be achieved by adding a sub-classifier network to the mutli-stream conv-LSTM. From frame t_{prev} to the current frame t, we have obtained target object's 2D location set (l_{in}, l_t) and the heatmap of frame t from the tracker. The selection model takes in (l_{in}, l_t) and the heatmap score of the current result, and compute a selection score, using a simple three-layers neural network s_φ with learnable parameters φ. Let s_h denotes the heatmap score, and v'_{in} represents the previous background motion, the selection score s_t at frame t can be obtained:

$$s_t = s_\varphi((l_{in}, l_t) + v'_{in}, s_h). \tag{5}$$

During training, the positive samples are obtained from ground truth labels. The negative sample is generated by adding a random drift displacement on positive samples. The average of the displacement is larger than the mean of the displacements in the previous t_{prev} frames. The loss function L_{select} for the trajectory classification task is a binary cross entropy:

$$L_{select} = \log(1 + \exp(-y_s s_t)), \tag{6}$$

where s_t is the selection score at frame t and $y_s \in \{1, 0\}$ is its ground-truth label.

In testing, our approach dynamically switches between the tracker and the trajectory prediction states by comparing their trajectory confidences s_t. When the target object suffers from occlusion from frame t, the final result will follow the trajectory predictions from $t+1$ to $t+t_{future}$.

4 Implementation details

Network Architecture For trajectory prediction model, we use a general seq-to-seq LSTM network [19, 20] as our backbone, where the encoder consists of 11 LSTM cells and the decoder consists of 5 LSTM cells. We modify each LSTM cell to conv-LSTM to handle the multi-channel feature maps. All the convolutional filter sizes are 3 × 3. A two streams of fully-convolutional network with 5 convolutional layers of stride 4 extracts features from the location map and image. The two streams feature maps are concatenated before encoding of LSTM. After decoding of LSTM , each response map representing the future location state is generated by three deconvolutional layers with stride 4. Finally, based on the target object's past 11 locations and the tracker's current heatmap score, the selection score is provided from a network of 3 1-D convolutional layers following a sigmoid activate function.

Training For the trajectory prediction network, we choose the successive 16 frames in a video as the temporal range of one sample. For each sample, we select the first 11 frames as the past states and regard the rest 5 frames as the future states. Thus, we use past 11 frames states to predict locations in next 5 frames. The response map and the location map are generated based on the target object's location by compensating the background motion. For the trajectory selection branch, we choose the target object's locations in first 12 frames from the the successive 16 frames as one sample. For each sample, we regard the last frame location as the future state. Specifically, we consider random translations (up to the mean of the displacements in the past frames). In training, we use the Adam optimization strategy with the learning rate of 0.0002. We train all our models using ImageNet-VID [25].

Inference In testing, our method needs the first 11 frames states, then is evaluated once per frame without any adaptation online. To deal with the long initialization problem, we repeat the first frame state to reach the 11 frames initialization quantity at the first 11 frames. It can be seen as the target object stays still at the initial location. Our trajectory obtains the motion information from the tracker's results.

	SiamRPN++	DCFST	ATOM	SiamMask	DIMP	**Ours-DiMP**
	[16]	[36]	[8]	[29]	[4]	
EAO ↑	0.285	0.361	0.292	0.287	0.305	0.316
Accuracy ↑	0.599	0.589	0.603	0.602	0.589	0.588
Robustness ↓	0.482	0.321	0.411	0.426	0.361	0.311

Table 1. State-of-the-art comparison on the VOT2019 dataset in terms of expected average overlap (EAO), accuracy and robustness.

	DaSiamRPN	ATOM	CCOT	MDNet	ECO	SiamRPN++	UPDT	DiMP	**Ours-DiMP**
	[37]	[8]	[11]	[23]	[9]	[16]	[5]	[4]	
OTB-100	65.8	66.9	68.2	67.8	69.1	69.6	**70.2**	67.7	69.3
UAV123	58.6	63.1	51.3	52.8	52.5	61.3	54.5	64.3	**64.9**

Table 2. State-of-the-art comparison on OTB-100 and UAV123 datasets in terms of area-under-the-curve (AUC) score.

5 Experiment Results

In this section, we evaluate our approach on VOT-2019[15], OTB-100[30] and UAV123[22] benchmarks. We choose the Siamese framework based tracker DIMP[4] and SiamMask[29] as our baseline trackers. On a single Nvidia GTX 1080Ti GPU, we achieve a tracking speed of 11 FPS when employing DIMP as the base tracker and 13 FPS for SiamMask.

5.1 Comparison with the state-of-the-art

VOT2019 Dataset[15]: We evaluate our approach on the 2019 version of Visual Object Tracking (VOT) consisting of 60 challenging videos. Following the evaluation protocol of VOT2019, we adopt the expected average overlap (EAO), accuracy (average overlap over successfully tracked frames) and robustness (failure rate) to compare different trackers. The detailed comparisons are reported in Table 1. Compared to DiMP, our approach has a 13% lower failure rate, while achieving similar accuracy. This shows that trajectory prediction is crucial for robust tracking.
OTB-100 Dataset[30]: Table 2 shows the AUC scores over all the 100 videos in the dataset. Among the compared methods, UPDT achieves the best results with an AUC score of 70.2%. Ours-DiMP achieves an AUC score of 69.3%, compared with our baseline tracker DiMP 67.7%.
UAV123 Dataset[22]: This dataset consists of 123 low altitude aerial videos captured from a UAV. Compared to other datasets, UAV123 has heavier camera motion that affects the target's trajectory severely. Results in terms of AUC are shown in Table 2. Our method, Ours-DiMP, achieves the best AUC score of 64.9%, verifying the strong trajectory prediction abilities of our tracker under the heavy camera motion.

5.2 Attributes Analysis

To analyze the performance on occlusion and other video attributes, we compare our method with the baseline tracker DiMP on OTB-100 and UAV123 datasets. Table 3 shows the AUC scores of all the 11 attributes in the OTB-100 dataset. Compared with DiMP, our method achieves a significant gain of 3.5% on the occlusion attribute. Specially, our method obtains a improvement of about 4% in AUC score on low resolution, background clutter, out of view attributes. The AUC scores of the 11 attributes in the UAV123 dataset are reported in Table 4. Our method outperforms DiMP with a relative gain of 1.3% on the occlusion attribute.

Since the occlusion and out of view attributes diminish the target object's appearance in the image frame, it is challenging for the appearance-based tracker to detect target in the current image. For rotation and fast motion situation that have little interference on object's appearance, our method obtains similar performance compared with DiMP's. These results demonstrate the effectiveness of our trajectory prediction.

	LR	BC	OV	OCC	MB	SV	DEF	IV	FM	OPR	IPR
DiMP[4]	58.4	62.7	60.2	63.5	69.0	67.8	65.8	68.5	67.7	66.7	68.5
Ours-DiMP	63.2	67.0	64.4	67.0	71.7	70.3	68.3	70.6	69.0	67.8	69.0

Table 3. Baseline tracker comparison on OTB-100 dataset in terms of AUC score on 11 attributes, including low resolution (LR), background clutters (BC), out-of-view (OV), occlusion (OCC), motion blur (MB), scale variation (SV), deformation (DEF), illumination variation (IV), fast motion (FM), out-of-plane rotation (OPR) and inplane rotation (IPR).

	LR	BC	OV	OCC	MB	SV	DEF	IV	FM	OPR	IPR
DiMP[4]	65.5	64.0	62.2	60.8	57.5	49.5	43.5	62.6	58.4	61.6	47.8
Ours-DiMP	66.6	64.8	63.5	62.1	58.9	50.6	43.5	63.3	59.7	61.5	47.1

Table 4. Baseline tracker comparison on UAV123 dataset in terms of AUC score on 11 attributes.

5.3 Ablation Studies

We perform an analysis of the proposed model prediction architecture. Experiments are performed on VOT2018[14] dataset.

Impact of Different Baseline Trackers Since the baseline tracker provide our method the heatmap score and tracking state, its performance is important to our method. To analyze the influence of baseline tracker, we choose another popular Siamese-framework based tracker SiamMask [29] as our baseline tracker. Compared with SiamMask, our method, namely Ours-SiamMask, improves the performance on all EAO, accuracy and robustness criteria. In particular, Ours-SiamMask obtains a significant relative gain of 3.3% in EAO, compared to SiamMask. Compared with the baseline variant tracker SiamMask-LD which improved by training on larger dataset, our corresponding, namely Ours-SiamMask-LD, achieves the gain of 4.73% and 14.1% in EAO and robustness respectively. Our method obtains a processing speed of 13 FPS which includes the processing time of the baseline tracker. The results for this analysis verifies the strong generalization abilities of our method, as shown in table 5.

	SiamMask [29]	SiamMask-LD [29]	**Ours-SiamMask**	**Ours-SiamMask-LD**
EAO ↑	0.380	0.422	0.398	**0.436**
Accuracy ↑	0.610	0.599	**0.616**	0.604
Robustness ↓	0.281	0.234	0.258	**0.201**
Speed ↑	**60**	43	13	13

Table 5. Analysis of different tracker models on the VOT2018 dataset in terms of EAO, accuracy and robustness.

Impact of Multi-Cues We make an ablation study to see how the background motion cue, the location map cue and image cue contributed overall tracking

	Tracker	No bg	No img	No loc	Temp-21	Temp-51	No Heatmap	Weight	Ours
EAO ↑	0.422	0.394	0.434	0.429	0.431	0.425	0.326	0.433	0.436

Table 6. Analysis of the impact of multi-cues, different temporal range of inputs and different selection mechanisms on the VOT2018 dataset.

performances respectively. We compare three different inputs. **No bg:** The trajectory prediction network predicts the target object's locations without background motion cue. Thus, the camera motion will affects the target object's trajectory. **No img:** Here, we use only the location map cue and the background motion cue. **No loc:** We utilize the target object's location value directly instead of the location map. The results are shown in table 6. **No bg** achieves an EAO score of 0.394, even worse than the baseline tracker. **No img**, which can exploit background information, provides a substantial improvement, achieving an AUC score of 0.434. This highlights the importance of employing the background motion prediction for compensating the target's trajectory. Our complete method outperforms **No loc** by 0.7%. This proves that the location map is a better way to represent the target object's trajectory in the image domain. Our complete method obtains the best results, which means each cue in our inputs is beneficial to improve performance.

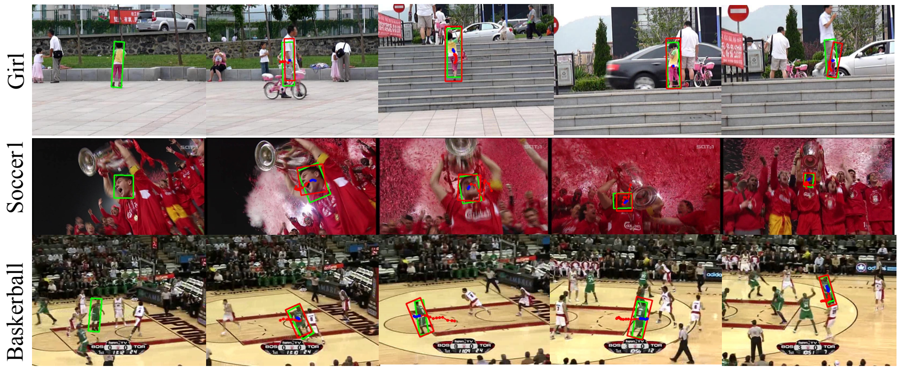

Fig. 5. Qualitative results of Ours-SiamMask for sequences from VOT2018. The red dots denote the target's past locations and the blue ones are our prediction. In comparison with the groundtruth (green bounding box), our method(red one) performs well under full occlusion in Girl and Soccer1 sequences.

Impact of Temporal Range We analyze the impact of the input's temporal range. Our basic idea is that using a short-term time window of one or two seconds for observation to predict the target object's future location. Thus, we choose the inputs' temporal range variants of 21 frames and 51 frames, namely

Temp-21 and Temp-51 respectively. Our complete method, namely Ours, takes 11 frames inputs, as mentioned in section 4. The results are reported in table 6. The Temp-21 variant outperforms the Temp-51 variant by 0.5%. Our complete method with the temporal range of 11 frames obtains the best performance. This indicates that the method with shorter length of the temporal range obtains better performance. This is due to the target motion is vary with time and also relative to the testing sequences.

Impact of Selection Mechanism We analyze the impact of trajectory selection mechanism by comparing three different variants. **No heatmap:** the selection score is evaluated based on the target trajectory without the heatmap score. **Weight:** The heatmap score is treated as a weight to the selection score instead of the trajectory selection network's input. **Ours:** A sub-classifier network takes in target's locations and corresponding heatmap score, and outputs a selection score, as described in section 3.3. The results are shown in table 6. **No heatmap** even makes the baseline tracker worse. In contrast, by considering the heatmap score from the tracker's appearance inference, our method obtains a significant gain of about 1.4% in EAO score over the baseline tracker. It also indicates that combing the heatmap score into a network is a more effective way than using it as a weight. These results demonstrate that our method can effectively switches between the tracker state and trajectory prediction.

6 Conclusions

We introduce a background motion model that captures the global background motion between adjacent frames to represent the effect of camera motion on image coordinates. This background motion is important to compensate the motion of the camera. We propose a new trajectory prediction model that learns from the target object's observations in several previous frames and predicts the locations of the target object in the subsequent future frames. A multi-stream conv-LSTM architecture is introduced to encode and decode temporal evolution in these observations. We also present a trajectory-guided tracking mechanism by using a trajectory selection score, which helps the tracker to switch dynamically between the current tracking status and our trajectory predictor, particularly when the target object is occluded.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016)
2. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: 2009 IEEE Conference on computer vision and Pattern Recognition. pp. 983–990. IEEE (2009)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)
4. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6182–6191 (2019)
5. Bhat, G., Johnander, J., Danelljan, M., Shahbaz Khan, F., Felsberg, M.: Unveiling the power of deep tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 483–498 (2018)
6. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2544–2550. IEEE (2010)
7. Cui, Z., Xiao, S., Feng, J., Yan, S.: Recurrently target-attending tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1449–1458 (2016)
8. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4660–4669 (2019)
9. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: Efficient convolution operators for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6638–6646 (2017)
10. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE international conference on computer vision. pp. 4310–4318 (2015)
11. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European conference on computer vision. pp. 472–488. Springer (2016)
12. Ellis, D., Sommerlade, E., Reid, I.: Modelling pedestrian trajectory patterns with gaussian processes. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. pp. 1229–1234. IEEE (2009)
13. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. Image and vision computing **14**(8), 609–615 (1996)
14. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al.: The sixth visual object tracking vot2018 challenge results. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
15. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Cehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., et al.: The seventh visual object tracking vot2019 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
16. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4282–4291 (2019)

17. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8971–8980 (2018)
18. Liang, J., Jiang, L., Niebles, J.C., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2019)
19. Liu, R., Bao, F., Gao, G., Zhang, H., Wang, Y.: Improving mongolian phrase break prediction by using syllable and morphological embeddings with bilstm model. In: Interspeech. pp. 57–61 (2018)
20. Liu, R., Bao, F., Gao, G., Zhang, H., Wang, Y.: A lstm approach with sub-word embeddings for mongolian phrase break prediction. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2448–2455 (2018)
21. Manh, H., Alaghband, G.: Scene-lstm: A model for human trajectory prediction. arXiv preprint arXiv:1808.04018 (2018)
22. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: European conference on computer vision. pp. 445–461. Springer (2016)
23. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4293–4302 (2016)
24. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
26. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 300–311 (2017)
27. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1349–1358 (2019)
28. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1442–1468 (2013)
29. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1328–1338 (2019)
30. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015)
31. Yagi, T., Mangalam, K., Yonetani, R., Sato, Y.: Future person localization in first-person videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7593–7602 (2018)
32. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing **74**(18), 3823–3831 (2011)
33. Yang, T., Chan, A.B.: Recurrent filter learning for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2010–2019 (2017)

34. Yang, T., Chan, A.B.: Learning dynamic memory networks for object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 152–167 (2018)
35. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) **38**(4), 13 (2006)
36. Zheng, L., Tang, M., Lu, H., et al.: Learning features with differentiable closed-form solver for tracking. arXiv preprint arXiv:1906.10414 (2019)
37. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 101–117 (2018)
38. Zou, H., Su, H., Song, S., Zhu, J.: Understanding human behaviors in crowds by imitating the decision-making process. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)