Tracking Beyond Detection: Learning a Global Response Map for End-to-End Multi-Object Tracking

Xingyu Wan, Jiakai Cao, Sanping Zhou, *Member, IEEE*, Jinjun Wang, and Nanning Zheng

***Abstract*— Most of the existing Multi-Object Tracking (MOT) approaches follow the Tracking-by-Detection and Data Association paradigm, in which objects are firstly detected and then associated in the tracking process. In recent years, deep neural network has been utilized to obtain more discriminative appearance features for cross-frame association, and noticeable performance improvement has been reported. On the other hand, the Tracking-by-Detection framework is yet not completely end-to-end, which leads to huge computation and limited performance especially in the inference (tracking) process. To address this problem, we present an effective end-to-end deep learning framework which can directly take image-sequence/video as input and output the located and tracked objects of learned types. Specifically, a novel global response network is learned to project multiple objects in the image-sequence/video into a continuous response map, and the trajectory of each tracked object can then be easily picked out. The overall process is similar to how a detector inputs an image and outputs the bounding boxes of each detected object. Experimental results based on the MOT16 and MOT17 benchmarks show that our proposed on-line tracker achieves state-of-the-art performance on several tracking metrics.**

***Index Terms*— Multi-object tracking, global response map, deep neural network.**

I. Introduction

MULTI-OBJECT Tracking (MOT) aims to use image measurements and predictive dynamic models to consistently estimate the states of multiple objects over discrete time steps corresponding to video frames. The major challenges of MOT are to continuously and effectively model the vast variety of objects with high uncertainty in arbitrary scenarios, caused by occlusions, illumination variations, motion blur, false alarm, etc., [1]. There are three key issues that a MOT framework should handle: 1) Modeling the dynamic motion of multiple objects; 2) Handling the entering/exiting of objects into/from the scene; 3) Robustness against occlusion and appearance/background variations. Single object tracking (SOT) [2]–[4] focus on 1) and 3) but simply applying multiple single object trackers for MOT task usually gives very limited performance due to 2).

With the significant progresses of object detection technologies, Tracking-by-Detection framework has become a leading paradigm whereby the detection results of objects are represented as bounding boxes (bboxes) in each frame and available as prior information. MOT is then casted as a problem of data association where the objective is to connect frame-based detection into trajectories across continuous frames using suitable measurements. The performance of these approaches largely depends on two key factors: Firstly the quality of detection results, where if the detection is missing or inaccurate at a single frame, or when occlusion occurs, the target state is then hard to estimate, and the target identity is prone to be lost; Secondly the data association model, where to achieve robust association across frames given the dynamic of objects, many works [5]–[8] conduct MOT in an off-line fashion with iterative solver [9] in order to make use of detection from both past and future, but is therefore time consuming and sensitive to the quality of appearance feature for association, not to mention scenarios where on-line processing is required.

More recently, many works have been proposed to utilize deep learning techniques to train Convolutional Neural Networks (CNNs) [10]–[12] from large scale datasets to obtain rich feature representations. These models have significantly improved the object detection performance and the quality of appearance feature. Several MOT approaches [13]–[15] have adopted Deep Neural Networks (DNNs) for association-based metric learning. They usually establish a robust motion model to predict the motion variations of targets and introduce a well-trained appearance model to extract deep features from region of interest (ROI) for image patches, and finally some similarity distances are adopted to measure the affinity of two ROIs for pair-wise association. Furthermore, aiming to learn a robust metric for feature representation, several works [16]–[18] take multiple features of objects in the scene by integrating a myriad of components such as motion, appearance, interaction, etc. Some works [19], [20] even consider to combine temporal components to analyze long-term variation by using Long

Manuscript received July 12, 2020; revised April 26, 2021; accepted August 30, 2021. Date of publication September 22, 2021; date of current version September 30, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700800, in part by the National Science Foundation of China under Grant 62088102 and Grant 6210020658, in part by China Postdoctoral Science Foundation under Grant 2020M683490, and in part the Youth Program of Shaanxi Natural Science Foundation under Grant 2021JQ-054. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mohammed Bennamoun. *(Corresponding author: Jinjun Wang.)*

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: jinjun@mail.xjtu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3113169, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3113169

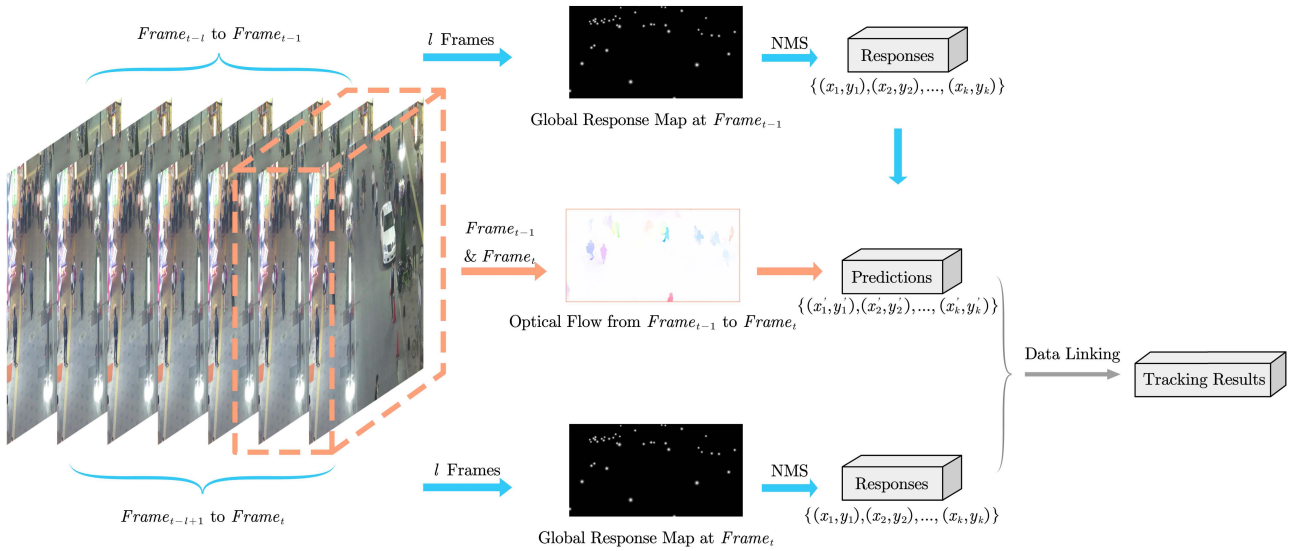

Fig. 1. The framework of our proposed MOT paradigm, composed of two modules, specifically the global object response generation module, and the motion displacement regression module respectively. Finally a simple data linking post-process is applied on the outputs of the two modules to generate the trajectory representation. The global response generator is designed to extract the attributes of "presence of object" and "x/y", and the motion displacement regressor is designed to retrieve the attribute of "$\Delta x/\Delta y$".

Short-Term Memory (LSTM). Since these methods are still based on disjoint detection/association steps, the computation is huge, and the performance is limited without end-to-end (*i.e.*, from image-sequence/video to trajectory) capacity. There are works that attempt end-to-end training for Tracking-by-Detection and association framework [21]–[23], but these approaches are still extended from image-based object detectors, the complete systems are not time-efficient and the trade-off between detection and association is hard to balance.

In this paper, we introduce an end-to-end framework for MOT. The challenge is to find a suitable representation that is capable to handle both issues 1), 2) and 3) in an on-line manner. Our idea is to employ a modified object salience model to generate a global response map to locate the presence of multiple objects, such that for issue 1), the motion of each object is implicitly modeled, and for 2) and 3), minor occlusions/entering/exiting within the window of frames can be robustly handled. The global response map has multiple channels where each channel models the response for different attributes to define the state space of all trajectories, such as "presence of object", "x/y", "$\Delta x/\Delta y$", as well as any additional attributes in the future, such that 1) the object detection step is only implicitly modeled in the tracking process, and the spatial information of a target is no longer a bounding box region of interest, but a Gaussian-like distribution from 0 to 1; and 2) the inference process does not need complicated assignment process but just a simple linking step to extract the multiple trajectories. In this paper, we introduced a logical inference methodology to estimate the actual state of target response based on the sequence of global response maps. Conceptually speaking, the sub-module for extracting each attribute of a state is similar to an ad-hoc network. For example, the attributes of "presence of object" and "x/y" are from a sub-module similar to an object locating network, the attribute of "$\Delta x/\Delta y$" is from a sub-module similar to an optical flow extraction network, etc. This can be further extended to include *width*, *height*, *orientation*, *depth*, etc., by adding suitable sub-modules in the future. The most important part is that, we are able to integrate all these sub-modules into one end-to-end network for MOT in one feed-forward step without any exhaustive cropping and iterating. The overall framework is illustrated in Fig. 1, and the main contributions of this work can be summarized as follows:

- We propose a novel representation schema and network structure to perform end-to-end MOT of learned object types. Different from existing Tracking-by-Detection and data association based approaches, our proposed method takes image-sequence/video as input and generates the trajectories of multiple objects in an end-to-end fashion, where multiple attributes that define the state space of each trajectory are obtainable from the global response map generated by our model.
- The proposed network includes a sub-network (*i.e.*, the global response generator) that operates like an image-sequence/video-based object locator and is capable of handling the occlusion problem. From within the defined time window, the module can still maintain a positive response even when target is occluded, and thus significantly reduce the false negatives.
- The proposed network also includes a sub-network (*i.e.*, the motion displacement regressor) that operates like an optical flow extraction network with a motion displacement regressor for estimating the motion dynamics. The module also helps solving the uncertain assignment problem in one single forward propagation.
- The proposed multi-object tracking network is complete end-to-end without any detection/appearance priors. The experimental results show that our tracker achieves superior performance over the state-of-the-art approaches on public benchmarks.

The following paragraphs of this paper are organized as follows: Section II reviews some related works. In Section III,

we introduce our method in detail, which includes the representation schema of encoding foreground objects on a global response generator, along with a logical inference approach for integrating temporal information to handle occlusion, and the methodology of using pixel-wise optical flow to train a motion regressor to help in the data linking post-processing. The experimental results are presented in Section IV, and the conclusion comes in Section V.

II. Related Work

A. Detector-Based Tracking

Owing to the rapid developments of object detection techniques, such as Faster R-CNN [24] and SDP [25], that obtains frame-based object detection results as initialization/priors, MOT can be conducted within Tracking-by-Detection paradigm where the objective is to connect detection outputs into trajectories across video frames using reasonable measurements, which therefore casts the MOT problem as global data association. Traditional data association techniques including the Multiple Hypothesis Tracker (MHT) [26] and Joint Probabilistic Data Association Filter (JPDAF) [27] aim to establish sophisticated models to capture the combinatorial complexity on a frame-by-frame basis. Aiming at global optimization with simplified models, the flow network formulations [6], [28]–[30] and probabilistic graphical models [13], [31]–[33] are considered, along with shortest-path, min-cost algorithms or even graph multi-cut formulations [34]. Most existing detector-based trackers highly rely on the quality of detection results, and to handle imperfect detections, several works [5]–[8] conduct MOT in off-line fashion to handle ambiguous tracking results for a robust tracking performance. Xu *et al.* [23] proposed to conduct data association process in an end-to-end fashion with a deep Hungarian network, which improved both the association efficiency and accuracy but a separate detection and embedding network is still needed for generating the cost matrix. Due to the two-stage nature with low processing speed, these detector-based methods are not applicable to real-time vision tasks. Compare to these methods, our proposed approach runs in an on-line manner without being bounded to specific object detection techniques and does not require complicated data association step.

B. Deep Metric Learning for Visual Tracking

Learning effective feature representations for targets of interest plays a central role in visual tracking. Most of the popular works for single object tracking (SOT) [2]–[4] adopt DNNs for robust feature representations. GOTURN [2] introduced a Siamese CNN to regress target bbox within a search region for real-time SOT. Zhang *et al.* [4] proposed a latent constrained correlation filter to sample the optimal target in a constrained subspace. Ding *et al.* [3] improved the kernelized correlation filters in SOT approaches to a quadrangle variant by regressing the four corners of target positions. Using DNNs for feature learning and correlation filters for target locating and bbox regressing established a leading paradigm for achieving promising tracking accuracy and running speed simultaneously in SOT task. However, simply applying such methodology for MOT task usually gives very limited performance due to the sophisticated hypothesis spaces for handling the entering/exiting of targets into/from the scene.

Metric learning based on DNNs for object appearance representation and computation of the affinity between measurements has become a popular trend in MOT community. Most trackers in this line of works model different features of objects from traditional CNNs [35], [36], Gabor CNNs [12], or even RNNs [18], [19], [37] by incorporating a myriad of components such as motion, appearance, interaction, social behavior, etc., Leal-Taixé *et al.* [35] adopted a Siamese CNN to learn local features from both RGB images and optical flow maps. Robicquet *et al.* [36] introduced social sensitivity to describe the interaction between two targets and used this definition to help data association. Later on, inspired by the success of Recurrent Neural Networks (RNNs) in language modeling [38], several works [18], [37] have attempted to learn an end-to-end representation for state estimation utilizing RNNs. Sadeghian *et al.* [18] proposed an off-line metric learning framework using a hierarchical RNN to encode long-term temporal dependencies across multiple cues, *i.e.*, appearance, motion and interaction. Milan *et al.* [37] presented an on-line RNN-based approach which is capable of performing prediction, data association and state update within a unified network. Followed by these, Wan *et al.* [19] extended the research of RNN-based methods and leveraged the power of Long Short-Term Memory (LSTM) for learning a discriminative model of object trajectory by integrating dynamic features both in temporal and spatial. For the on-line MOT task, these methods may not perform well when heavy occlusion or misdetection downgrades the robustness of appearance model. Differently, in our work, the occlusion problem is well handled by considering a window of frames to locate objects with an introduced logical inference methodology, without explicit appearance feature for metric learning.

III. Our Proposed MOT Algorithm

Aiming to join the separate procedures of object detection and data association for better tracking performance, here we introduce an end-to-end MOT framework as shown in Fig. 1. We take consecutive image frames as network inputs to learn 1) a global response map to locate interested targets globally as well as the object attributes, such that, the location of each target and other attributes can be easily obtained in a post-processing step with the help of a simple Non-Maximum Suppression (NMS), and 2) the motion displacements for these targets from a frame-wise optical-flow-like offset, with which after the post-processing step the "$\Delta x/\Delta y$" attributes of each objects can be also obtained, and thus completing the image-sequence/video to multiple trajectories process.

Our proposed MOT algorithm is presented as follows. The first section talks about the global object response generation sub-network, the next section is the motion displacement regression sub-network, and the final section is the post-processing step to obtain the trajectory representation based on the obtained attributes.

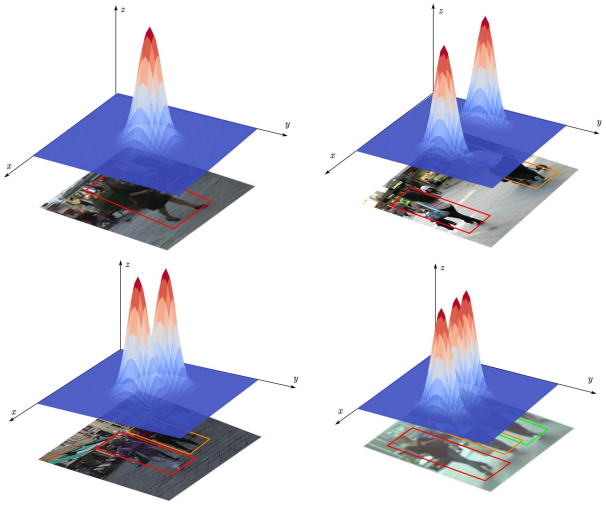

Fig. 2. An exemplary of local response maps viewed in 3D. Each response with a Gaussian-like distribution represents a foreground object to be tracked.

A. Object Locating Using Global Response Map

The goal of tracking is to consistently maintain the estimation of object states over discrete time step. In this specific computer vision task, using a well-trained class-specific detector to filter out all the regions of interest over the image frame may not be necessary. Here we propose a simpler and more efficient way to locate objects for MOT. For all the targets to be tracked at each time step, we take them all as foreground objects and represent them as Gaussian-like distributions from 0 to 1 with peak value at their center points on a saliency map. As shown in Fig. 2, on this response map, each Gaussian-like distribution represents a foreground object to be tracked, the x, y coordinates correspond to the object spatial location, and z is a value from 0 to 1 corresponds to the actual status of object at current time step. Followed by CornerNet [39], we adopt the same object size-adaptive standard deviation for generating the kernel of our Gaussian-like distributions. Given a bounding box scale w and h, we first represent the kernel radius r as a size-adaptive quadratic parabolic equation,

$$\gamma_1 r^2 + \gamma_2 r + \gamma_3 = 0. \tag{1}$$

The coefficients of Eq. (1) are all related to the bounding box scales of ground-truth objects. Followed by this, the radius r and sigma σ of each Gaussian-like distribution are derived as:

$$r = \min_{i=1}^{3} \left| \frac{a_i + \sqrt{a_i^2 - b_i}}{2} \right|, \qquad \sigma = \frac{r}{3}. \tag{2}$$

where

$$a_i = \gamma_2 = \begin{cases} h + w, & i = 1 \\ 2 \times (h + w), & i = 2 \\ -2 \times (h + w), & i = 3. \end{cases} \tag{3}$$

$$b_i = 4\gamma_1\gamma_3 = \begin{cases} 4 \times \dfrac{h \times w \times (1 - \alpha)}{1 + \alpha}, & i = 1 \\ 16 \times h \times w \times (1 - \alpha), & i = 2 \\ 16 \times h \times w \times \alpha \times (\alpha - 1), & i = 3. \end{cases} \tag{4}$$

Here h and w denote the height and width of target bbox obtained from the ground-truth (GT) of training data, α is an invariant parameter denotes the minimum overlap with GT bboxes which we set to 0.7 in this work. Parameters a_i and b_i are derived from the coefficients $\gamma_1, \gamma_2, \gamma_3$ of the object size-adaptive quadratic parabolic equation with three different cases. We adopt the minimum value of $\{r_i, i = 1, 2, 3\}$ as the radius r of our Gaussian kernel, and the sigma σ is set to $\frac{r}{3}$ accordingly. In this way, our Gaussian-like distribution has a positive correlation with target size w and h.

1) Response Map Learning Network: The above representation as global response map is able to describe the object spatial location and actual state at the same time. To learn such representation, our tracking algorithm employs a HED-based [40] saliency detection network modified from [41]. Specifically, our object locating sub-network is an Auto-Encoder, which takes a time window of frames with a length l as inputs, and outputs a single channel response map after a $sigmoid$ function. We adopt the short connection strategy as [41] but remove the fusion layer, and we compute the average value of $1, 2, 3, 6$ side outputs as our network output before activation. Given a training sequence $X_l = \{I^{t-l}, \ldots, I^i, \ldots, I^t, I^i \in \mathbb{R}^{3*h*w}\}$, and label response map $Y \in \mathbb{R}^{1*h*w} = \{y_j, j = 1, \ldots, |Y|\}$, the standard cross entropy loss function for our network is given by

$$L(X_l, Y) = -\sum_{j=1}^{|Y|} (y_j log\mathbf{P}(y_j = 1|X_l) + (1 - y_j) log\mathbf{P}(y_j = 0|X_l)), \tag{5}$$

where $\mathbf{P}(y_j = 1|X_l)$ denotes the probability of the activation value at location j, and label Y is obtained using the following logical inference methodology from GT data.

2) Logical Inference Methodology for Handling Occlusion: When a target being tracked is unseen at a particular time step, that does not mean this target actually leaves the surveillance scene. In this work, we argue that the actual state of object presence in visual tracking scenes should be distinguished from image-based detection results, and this estimation of actual state can be learned using history priors. As described above, we use a $0/1$ response value to represent the target actual state at each time step. This representation should be estimated using not only image at current frame, but also images from the past. Here we introduce a logical inference methodology for estimating response value of each target as shown in Fig. 3. For target trajectories $\{T_j, j = 1, 2, \ldots m\}$ from GT, the response value z_j^t of target actual state at frame t is estimated upon a time window of past states $\{z_j^i, i = t-l, t-l+1, \ldots, t-1\}$ with a length l. The specific estimation method is described as follow,

$$z_j^t = \begin{cases} 1, & \text{if } z_j^{t-1} = 1 \text{ or } \frac{\sum_{i=t-l}^{i=t-1} z_j^i}{l} >= \beta \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here parameter β describes the proportion of positive states within the time window l. In general, if a target appeared to be existing during most of the past time, we take it still exist at current time step even when we got a negative from

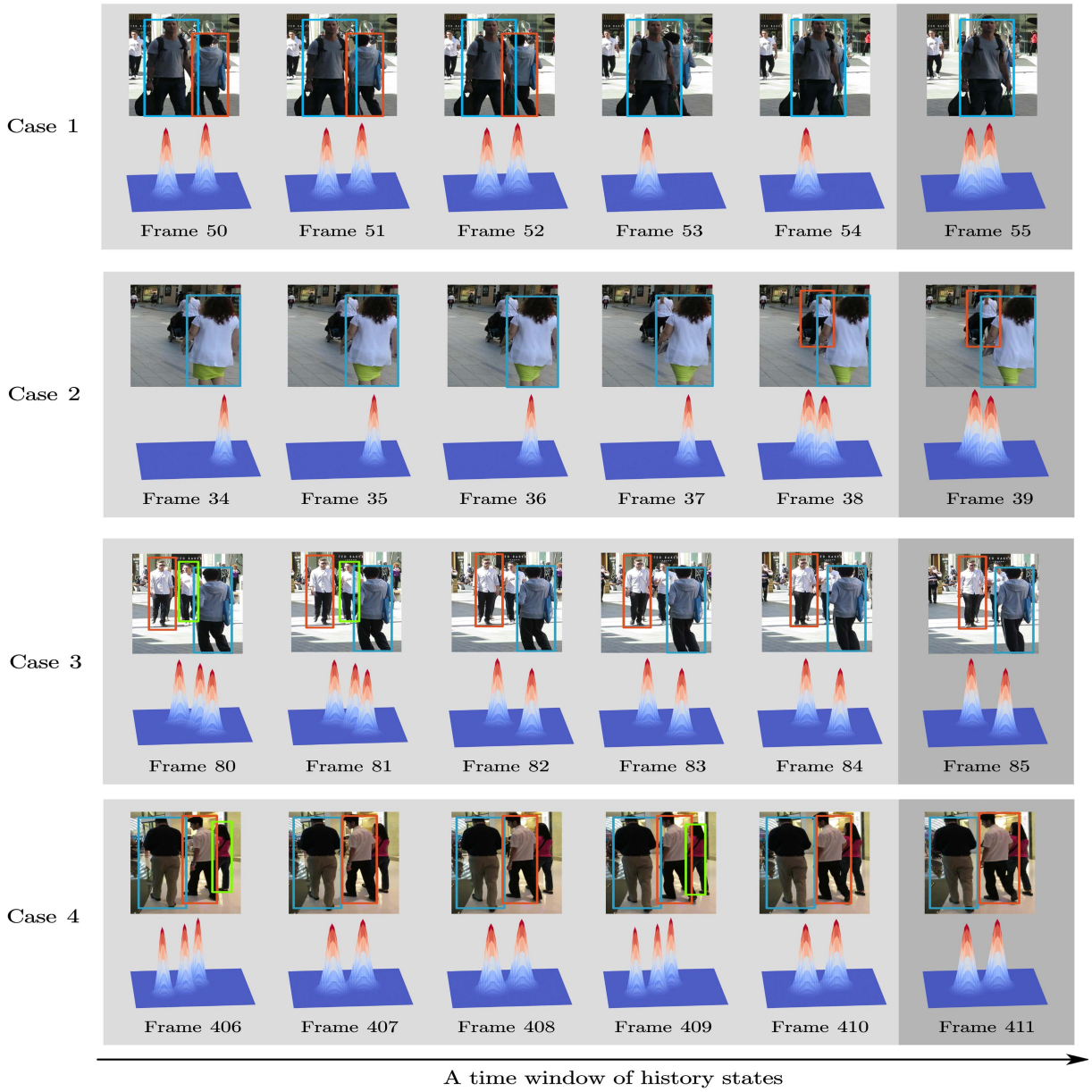

Fig. 3. An exemplary of our logical inference methodology for target response estimation (Better viewed in color). For each case, the first row is object detection results from MOT benchmark, and the second row is the corresponding response maps represent the actual states of targets. Four different cases are shown in this figure, the actual state of each observed object (annotated by blue, orange and green bounding boxes) at current frame (the last column) is estimated by a time window of history states.

image-based detector. Conversely, if a target remained negative during most of the past time, we consider it actually leaves the scene and take the positive detection as a false alarm. In addition, if target has a positive response at last time step, *i.e.*, $z_j^{t-1} = 1$, we set $z_j^t = 1$ accordingly no matter what the history states are. At training phase, this estimation strategy is employed to generate positive/negative samples from training data. By using this strategy, we can consistently maintain the correct positive responses for targets even when suffering occlusions.

B. Motion Displacement Regression

To obtain the predicted motion displacements of all located targets, most popular approaches iteratively cropped the image patch of each region of interest (ROI) to learn a regressed motion prediction. These approaches are time consuming and hard to achieve end-to-end training/inference.

In our proposed MOT algorithm, the motion dynamics are estimated from a motion displacement regression sub-network using a frame-wise optical-flow-like offset. As proposed in [42], we adopt FlowNet2 for our frame-wise optical flow extraction. Given two adjacent frames I^{t-1} and I^t, the optical flow estimation from frame $t-1$ to frame t can be derived as $W^{t-1} = \sum_i (u_i, v_i)$, $W^{t-1} \in \mathbb{R}^{2*h*w}$, where i denotes each pixel on the flow. After deriving the pixel-wise displacements from optical flow, we introduce a regression sub-network to learn the predicted motion displacements globally for all responses with Gaussian-like distributions.

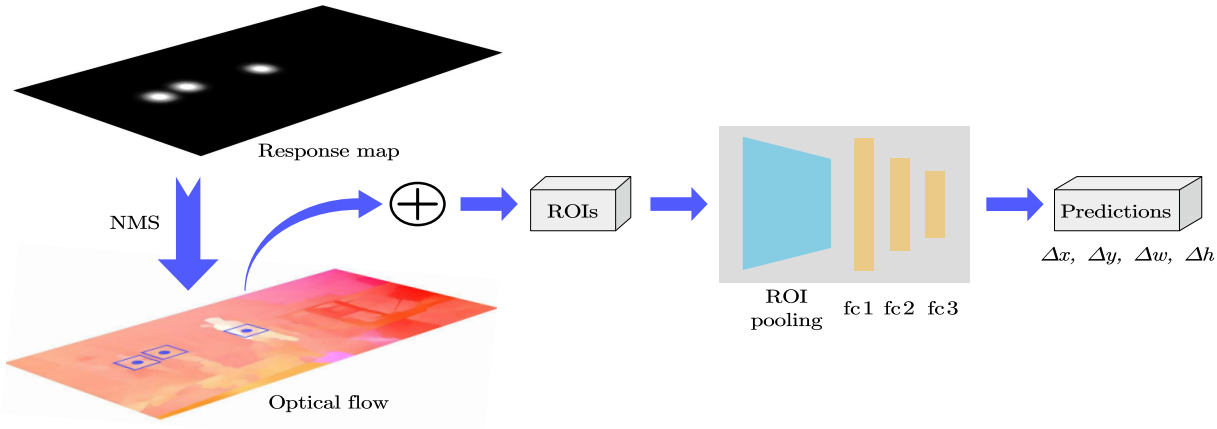

Fig. 4. The network structure of motion displacement regression. The inputs of this network are the center point distributions of located objects retrieved from global response map after NMS. The outputs are predicted motion displacements of all positive responses.

As shown in Fig. 4, we first conduct local NMS with a kernel size s on response map Z^t at frame t with a threshold value of response $Score$ to filter out the top k positive responses and retrieve their center point locations. For each retrieved response distribution with a center point (cx, cy) and fixed kernel size r_z, we take it as a region of interest (ROI), and sample all the displacements of ROIs from optical flow at the same time to obtain a concatenated featuremap $F^{t-1} \in \mathbb{R}^{k*2*r_z*r_z}$ for regression. The regression sub-network is composed of a ROI pooling [43] layer and several fully-connected layers. This network structure is designed to learn one accurate displacement value of $\{cx, cy, w, h\}$ for response point distribution from a ROI. The network output $D^t = \{d_j = (\Delta cx, \Delta cy, \Delta w, \Delta h), j = 1, \ldots, k\}$ is a movement displacement vector of all response points from frame $t-1$ to frame t. Given the ground-truth G^t and network output D^t, the loss function of our regression network is defined as

$$smoothL_1(G^t, D^t) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{7}$$

where x denotes the L1 loss between G^t and D^t. Incorporating ROI pooling into our regression network enables our tracker to estimate frame-wise motion displacements of all the observed targets in one single forward propagation without any cropping and network iteration.

C. Post-Processing for Trajectory Extraction

After the main tracking phase, the global response map Z^t and motion displacement estimation D^t are already obtained after one forward propagation using the proposed network. Next, similar to how an one-stage object detector does, such as YOLO [44], to extract the bbox of each detected object from some sort of object response map, mostly with the help of the NMS step, our tracker also needs one last step to convert the output maps into trajectory coordinate/attribute. We call this post-processing step as data linking stage.

Our data linking strategy is basically a sampling and matching process. We first obtain the predicted location map $P^t = (cx', cy', w', h')$ by adding the regressed motion displacement D^t to the previous target location. After generating Z^t and P^t, we directly extract the located values and predicted values for each corresponding target from two maps, and make assignments globally. Here we introduce two different matching approaches for generating trajectories.

1) Target-Independent Matching: As each windowed input from $t-N$ to $t-1$ generates response for t, we could simply project attributes "x/y" back with attributes "$\Delta x/\Delta y$" into the last response, and use a simple greedy matching algorithm based on center point distance to connect objects. Each object is matched with an observed response distribution at current frame with the minimum center point distance from last response, and unmatched tracks are immediately terminated after one iteration. In this way, we could extract the complete trajectory and represent it as a sequence of object attributes.

2) Target-Dependent Matching: It is observed that the assignment of objects may not be optimal with the above-mentioned greedy matching. Although the fundamental solution to improve the post-processing is to improve the quality of global response, we could also consider global assignment in the post-processing. Hence, for target-dependent matching, we compute a global cost matrix between predictions P^t and observations Z^t using the intersection-over-union (IOU) distance, then we solve the assignment problem optimally by minimize the global cost matrix. Specifically, for all tracks $T^{t-1}_{i=1,\ldots,N} = \{cx^i, cy^i, w^i, h^i\}$ at frame $t-1$, we first extract the predicted locations $P^t_{i=1,\ldots,N} \in P^t$ in state space at frame t from prediction map P^t. In observation space, the presence attribute of each target at frame t is extracted from the corresponding channel of learned response map Z^t. Then for all observed targets $Z^t_{j=1,\ldots,M} \in Z^t$ with a positive response value higher than $Score$, we extract the target locations from the motion channels of response map. After that, we compute the IOU distance between the P^t_i and Z^t_j to generate a $N*M$ dimensional cost matrix. Followed by this, the assignment problem leads to an optimal association between responses and tracks which can be solved by applying the Hungarian algorithm [45] to maximize the sum of all IOUs at frame t. After this, we adopt a simple matching cascade strategy for all unmatched responses and tracks as a refinement. Specifically, we set a constant parameter A_{max} representing the maximum age. For each response z^t_k not assigned to an existing track

Fig. 5. Qualitative results of our object locating method (column 1) and two popular detectors (Faster-RCNN [24](column 2) and SDP [25](column 3) respectively) on some challenging cases, best viewed in color. Our object response can still be located on the object even when occluded, which significantly helps the following data linking module to obtain a complete trajectory for the object.

at current frame, we compute the IOU distance between this response with each terminated track whose last frame is within a time window from $t-1$ to $t-A_{max}$. This computation is iteratively conducted frame-by-frame until the IOU distance is higher than IOU_{min}, and we take this response z_k^t to update the target state of corresponding terminated track as a match. After this matching cascade, all responses not assigned to an existing track will be initialized as a new track, and all tracks without an assigned response will be terminated.

IV. Experiment

A. Experimental Details

We report the performance of the proposed MOT algorithm on MOT16 and MOT17 benchmarks [46]. The provided detection results of MOT16 is DPM [47]. The MOT17 dataset contains the same video sequences(7 fully annotated training sequences and 7 testing sequences) as MOT16 but with two more sets of public detection results from Faster-RCNN [24] and SDP [25] respectively. We implemented our framework in Python3.6 using PyTorch, with six cores of 2.4GHz Inter Core E5-2680 and two NVIDIA GTX 1080Ti GPUs.

For object locating, we set time window length l to 20, and parameter β of logical estimation methodology to 0.6. At training phase, we used VGGNet [48] pre-trained on ImageNet [49] as the shared convolution blocks. The input of our network is a sequence of image frames which were resized to 512×960 for height and width. The total training epochs were 300 and the learning rate was initialized to $1e-3$ and divided by 10 every 100 epochs.

For motion displacement regression, we trained our model on 7 MOT training sequences with provided GT data. Here we set local NMS kernel size s to 3 and the threshold value of response $Score$ to 0.05, the parameter k for maximum number of positives to 60, and the kernel size r_z of ROI Pooling layer to 20. At training phase, we used FlowNet2 [42] pre-trained on MPI-Sintel [50] for extracting frame-wise optical flow. The input of our network for motion variation regression is a sequence of adjacent image frames which were resized to 1024×1920 for height and width. We set the initial learning rate to $1e-4$, the total training epochs to 500 and divided the learning rate by 2 at epoch 166, 250, 333 and 416 respectively. We used Adam [51] optimizer for both sub-networks. The parameters IOU_{min} and A_{max} for data linking were set to 0.7 and 10 respectively.

B. Object Locating Performance

In order to exam the validity of our proposed representation schema, we visualized the results of our object locating sub-network along with two popular detectors on some challenging sequences from MOT benchmark [46]. As shown in Fig. 5, compared with Faster-RCNN [24] and SDP [25] at challenging situations like small scales (row 1) and crowded (row 2), our object locating method (column 1) gives more accurate estimations centered on the foreground objects. Furthermore, at extremely occluded scenario (row 3), benefit from logical inference on history states, our method can still maintain positive and accurate responses for objects completely occluded at

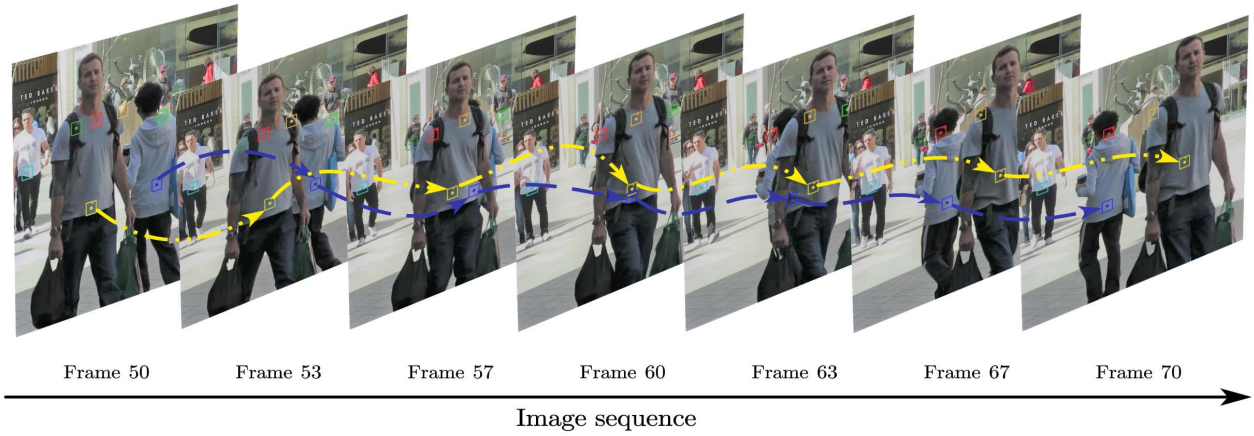

Fig. 6. An illustration of tracking results visualization using tracking as response points method on an extremely occluded scene. The kernel size r_z here is set to 20. Our tracker only takes raw image-sequence/video as inputs, and tracks on each foreground response distribution.

TABLE I
RUN-TIME COMPARISON AMONG OUR PROPOSED OBJECT LOCATOR AND SEVERAL POPULAR DETECTORS

Method	Year	ms/frame	FPS
Fastest-DPM [52]	2014	66	15
Fast-RCNN [43]	2015	320	3
Faster-RCNN [24]	2015	198	5
YOLO [44]	2016	22	45
SSD300 [53]	2016	17	58
Response Map (**Ours**)	2020	**5**	**200**

this frame, while detector-based trackers have to make further analyses on such case with complicated strategies.

In addition to the accuracy term, our object locating method runs much faster than DNN-based detectors. In Table I, we compared the running time of our proposed object locating method with several popular detectors. Here we report the running times of classic DPM detector with a speeding up version [52], two-stage CNN-based detectors Fast-RCNN of VGG-16 version [43] and Faster-RCNN with VGG-16 for both proposal and detection [24], one-stage CNN-based detectors YOLO using VGG-16 [44] and SSD with 300×300 image size [53]. Our method were measured by computing the average run-time on 7 test sequences from MOT benchmark [46]. The FPS of our method for one feed-forward propagation is 40 times faster than classic Faster-RCNN [24] and about $3-4$ times faster than real-time detectors YOLO [44] and SSD [53]. These evaluation results confirm that object locating from the global response map is faster and more accurate. More importantly, different from Tracking-by-Detection and association based methods, this object locator is utilized as a sub-network within our framework in an end-to-end fashion.

C. Tracking Performance Evaluation

1) Tracking as Response Points: Different from traditional Tracking-by-Detection paradigm, our proposed MOT method operates without using any detection results. Hence, instead of using the provided bounding box coordinates, our tracker only takes raw image-sequence/video as inputs, and tracks on each foreground response distribution. The representation of this tracking response points includes a center point coordinates (x, y) and a fixed kernel size r_z for each target. The exemplary visualization of our tracking as response points method is illustrated in Fig. 6. Owing to the accurate center point locations retrieved from global response map and effective motion model, our tracker is rather robust for maintaining the identities of tracked targets during short-term occlusions.

2) Comparison With State-of-the-Art Trackers: Since the existing evaluation metrics for MOT are designed for traditional Tracking-by-Detection paradigm where the tracking results are provided and expressed as bbox coordinates, in order to compare our tracker with other tracking methods, we ran one additional step that maps our generated response points to the provided detection results from MOT benchmarks [46] so that the required metrics can then be calculated. Specifically, we compute the cost matrix between response distributions from response map and public detections from MOT benchmarks [46] using center point distance at each time step. Then we make global optimal assignments same as section III$-C$. We take the retrieved positive responses from object locating network as priors. For those detections from MOT dataset not assigned to any positive response, we take them as false alarms. For those responses not assigned to any provided detection, we maintain the object location with an initialized bbox scale, and this scale is adjusted by motion displacement regression sub-network during tracking.

We evaluate our tracker on the test sets of both MOT16 and MOT17 benchmark. The evaluation is carried out according to the metrics used by the MOT benchmarks [67]–[69], which includes Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), ID F1 Score (IDF1), Mostly tracked targets (MT), Mostly lost targets (ML), the total number of false positives (FP), the total number of false negatives (FN), the total number of identity switches (ID Sw.), and the total number of times a trajectory is fragmented (Frag). Among these measurements, main evaluation metric MOTA is a combination of false positives, missed targets and identity switches, IDF1 computes the ratio of correctly identified detections over the average number of ground-truth and computed detections.

TABLE II

EVALUATION RESULTS ON MOT16 DATASET, COMPARED WITH CURRENT STATE-OF-THE-ART METHODS. THE BEST RESULTS ARE SHOWN IN **BOLD**. MODE WITH ○/× DENOTES AN ONLINE/OFFLINE METHOD, ↑ MEANS THE HIGHER THE BETTER, AND ↓ MEANS THE LOWER THE BETTER

Method	Mode	Year	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓	Frag ↓
STRN16 [54]	×	2019	48.5	73.7	53.9	17.0%	34.9%	9,038	84,178	747	2,919
LMP [34]	○	2019	48.8	**79.0**	51.3	18.2%	40.1%	6,654	86,245	481	595
KCF16 [55]	○	2019	48.8	75.7	47.2	15.8%	38.1%	5,875	86,567	906	1,116
AFN [56]	×	2018	49.0	78.0	48.2	19.1%	35.7%	9,508	82,506	899	1,383
eTC [57]	×	2019	49.2	75.5	56.1	17.3 %	40.3%	8,400	83,702	606	882
LSST16 [58]	○	2019	49.2	74.0	56.5	13.4%	41.4%	7,187	84,875	606	2,497
HCC [59]	○	2018	49.3	79.0	50.7	17.8%	39.9%	5,333	86,795	**391**	**535**
NOTA [60]	○	2019	49.8	74.5	55.3	17.9%	37.7%	7,248	83,614	614	1,372
Tracktor++v2 [61]	○	2019	56.2	79.2	54.9	20.7%	35.8%	**2,394**	76,844	617	1068
Ours (target-independent matching)	○	2020	59.9	73.4	49.5	**40.3%**	**19.8%**	21,243	50, 520	1,382	2,048
Ours (target-dependent matching)	○	2020	**62.4**	73.6	**64.0**	37.5%	20.7%	17,523	**50,155**	883	2,002

TABLE III

EVALUATION RESULTS ON MOT17 DATASET

Method	Mode	Year	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓	Frag ↓
STRN17 [54]	×	2019	50.9	75.6	56.0	18.9%	33.8%	25,295	249,365	2,397	9,363
jCC [62]	×	2018	51.2	75.9	54.5	20.9%	37.0%	25,937	247,822	1,802	2,984
NOTA [60]	○	2019	51.3	76.7	54.5	17.1%	35.4%	20,148	252,531	2,285	5,798
FWT [63]	○	2018	51.3	77.0	47.6	21.4%	35.2%	24,101	247,921	2,648	4,279
AFN17 [56]	×	2018	51.5	77.6	46.9	20.6%	35.5%	22,391	248,420	2,593	4,308
eHAF17 [64]	×	2018	51.8	77.0	54.7	23.4%	37.9%	33,212	236,772	1,834	**2,739**
eTC17 [57]	×	2019	51.9	76.3	58.1	23.1%	35.5%	36,164	232,783	2,288	3,071
FAMNet [65]	×	2019	52.0	76.5	48.7	19.1%	33.3%	14,138	253,616	3,072	5,318
JBNOT [66]	○	2019	52.6	77.1	50.8	19.7%	35.8%	31,572	232,659	3,050	3,792
LSST17 [58]	○	2019	54.7	75.9	62.3	20.4%	40.1%	26,091	228,434	**1,243**	3,726
Tracktor++v2 [61]	○	2019	56.3	**78.8**	55.1	21.1%	35.3%	**8,866**	235,449	1,987	3,763
Ours (target-independent matching)	○	2020	59.4	74.8	54.1	**40.2%**	**21.0%**	38,522	188, 736	4,111	7,538
Ours (target-dependent matching)	○	2020	**61.8**	75.5	**63.2**	36.7%	22.0%	35,168	**179,986**	3,589	7,640

Table II and Table III present the evaluation results on MOT16 and MOT17 dataset [46] respectively, in comparison with some of existing best performing peer-reviewed trackers both online and offline. From Table II and Table III we can see that the main evaluation metric MOTA of our proposed method surpasses all the other State-of-the-Art (SOTA) trackers. For instance, compared with the SOTA method tracktor++ [61] which conducts MOT with a modified object detector, our tracker gains 5.5 and 6.2 points of MOTA on MOT17 and MOT16 separately. Moreover, our tracker also achieves the best performance in IDF1, MT, ML and FN. Among these metrics, the highest MT and the lowest ML indicate the robustness of our tracker for maintaining targets identities.

Furthermore, the lowest FN of our tracker confirms that introducing object locating network to estimate the actual states does make an effort for reducing the false negatives caused by image-based detectors, and the occlusion problem is handled rather well. Meanwhile, the raising numbers of our FP mainly come from the inaccurate object scales provided by public detections from MOT benchmark, which were adopted as initialization in our tracker. More analyses on FP and FN terms can be found in section IV−*D*.

3) Run-Time Efficiency: At online inference phase, the run-times of each component tested on MOT benchmark is listed in Table IV. Our tracking as response points method outputs trajectories for all responses in one feed-forward propagation, and the total run-time is 162.5*ms* (6 FPS). This is noticeably faster than many "real-time" and online methods where "tracking" is conducted after a non-negligible detection step, such as those trackers [15], [70]–[73] based on Faster-RCNN [24] detectors as shown in Table V.

TABLE IV

TIMING OF EACH COMPONENT OF OUR PROPOSED NETWORK

Component	ms/frame	FPS
Object locating (including NMS)	108	9
Motion displacement regression	7.5	133
Data linking	47	21
Total	162.5	6

TABLE V

RUN-TIME (MS/FRAME) COMPARISON ON MOT16 BENCHMARK

Method	Detection	Tracking	Total	FPS
RAR16wVGG [70]	198	>600	>798	<1.5
POI [71]	198	>160	>358	<3
CNNMTT [72]	198	>150	>348	<3
TAP [73]	198	>120	>318	<3.5
Deep SORT [15]	198	>25	>223	<4.5
ours	**0**	**162.5**	**162.5**	**6**

D. Ablation Study

The ablation study is designed to further analyze the effectiveness of each module of our proposed tracker and their corresponding impacts on tracking performance. For our object response generator, we analyzed the terms of input length of history priors, hyper-parameter settings and the NMS process individually. For our motion regressor, we compared different experimental settings of optical flow extraction and ROI pooling processes. And at last we compared two data linking strategies for robustness validation.

1) History Priors for Object Locating: Our network encodes a sequence of raw image frames inputs into a global response map using a proposed logical inference process. Table VI shows the ablation study on history priors, which is

TABLE VI

ABLATION STUDY ON DIFFERENT HISTORY PRIORS

Input frames	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
l =5	62.0	37.7%	20.7%	18308	50039	909
l =10	62.0	37.2%	22.8%	17908	49869	914
l =20	**62.4**	37.5%	20.7%	17523	50155	**883**
l =30	62.1	**41.6%**	**20.3%**	18777	**49242**	1020
w.o.history	57.5	29.4%	26.5%	**13401**	71241	899

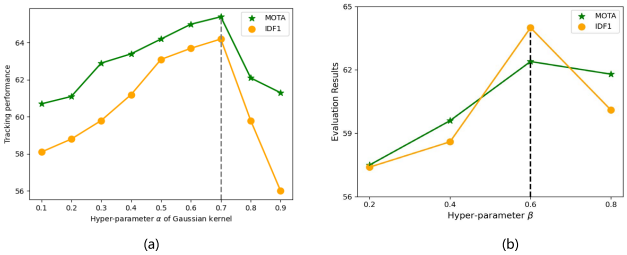

Fig. 7. Analysis on hyper-parameter settings of (a) minimum overlap α for Gaussian kernel generation, and (b) positive proportion β for logical inference.

intended to demonstrate the superiority of our spatial-temporal response generator based on history inference. The last row of table VI presents the tracking result of our method when using one single image as network input. Without engaging any history information, our tracker still maintains a competitive tracking performance (MOTA 57.5) with other SOTA methods. This proves the effectiveness of our object locator in spatial space and the accuracy of our motion regression sub-network. Our network conducts hypothesis inference spatially on single image frame when no history prior can be used, thus a negative value is directly output when a foreground object is unseen at current frame. This can explain the raising numbers of FN, and without any hypothesis during occlusion, the FP is also decreased at the same time. When we use more history priors as network inputs, our object locator works more accurate by learning temporal information and thus yields better tracking performance. The improvements mainly reflect on FP and FN, which indicates the hypothesis inference process during occlusion is more accurate when more temporal information are learned. We obtained the best performance when we set l to 20 in our experiments.

Moreover, from the experiments we also observed that the more image frames we used as inputs, the more space complexity we will gain for training our object locator. For instance, the GPU memory for training our object locator without history prior was 5329 MB. When we set l to 5, the GPU memory usage was 6151 MB. When we increased l to 20, the GPU memory usage was 12991 MB, which is more than twice of the space complexity when $l = 5$.

2) Hyper-Parameter Settings: We analyzed the impacts of using different hyper-parameters α and β on main evaluation metrics MOTA and IDF1 scores as show in Fig. 7. The optimal Gaussian kernel radius r of each foreground distribution is determined by ensuring a pair of points within this Gaussian radius could generate a bbox with at least α IOU with GT annotation. We conducted the comparative analysis for generating different Gaussian-like distributions with different α as illustrated in Figure 7 (a). Lower hyper-parameter α results in larger Gaussian kernel radius r, while higher α results in smaller radius r in general. Large radius r ($\alpha < 0.4$) may cause two major problems: 1) multiple distributions are largely overlapped when targets are too close; 2) uneven distributions of hard/easy training samples. Small radius ($\alpha > 0.7$) can easily generate uneven distributions of positive/negative samples, which makes the positive signals hard to be recognized for our network. The evaluation results in in Figure 7 (a) indicate that medium α (0.5-0.7) can provide us promising tracking performances, and we obtained the best performance when α was set to 0.7 in our experiments.

TABLE VII

EVALUATION RESULTS OF OUR METHOD ON MOT16 DATASET WITH DIFFERENT EXPERIMENTAL SETTINGS

Method	MOTA↑	MT↑	ML↓	FP↓	FN↓	ID Sw.↓
baseline	**62.4**	37.5%	20.7%	17523	**50155**	883
NMS 10	61.1	36.0%	21.2%	13607	51701	636
NMS 20	57.6	32.5%	26.5%	16253	59943	1032
NMS 30	56.4	26.0%	27.7%	**12635**	65953	871
interval 5	61.4	37.1%	18.2%	18004	56667	**781**
interval 10	61.5	35.8%	20.0%	13643	53467	3041
interval 20	60.8	32.8%	22.1%	17365	55024	3928
ROI 10	61.5	35.2%	21.1%	12476	54974	2715
ROI 30	62.0	38.2%	23.7%	17410	50986	911
ROI 40	61.5	34.3%	22.7%	17635	55950	3090
w.o. Hungarian	59.9	**40.3%**	**19.8%**	21243	50520	1382

In our proposed logical inference methodology, hyper-parameter β of Eq.(6) is used for estimating response value of each target on global response map. Increasing β results in generating less positive target-wise distributions on response map, while decreasing β results in bringing more positives on the contrary. Introducing more correct positive distributions helps in reducing the False Negatives, while introducing wrong ones will increase the False Positives. We obtained the best performance when β is set to 0.6 in our experiments.

3) Kernel Size for Local NMS: After one forward propagation of our network, we conduct local NMS to retrieve positive responses from the output global response map. We further investigated the aspect of NMS kernel size as shown in table VII and Fig. 8 (a). Compared to the baseline method (NMS kernel size is 3×3), the main evaluation metric MOTA decreases when we increase the NMS kernel size gradually. The worse MT and ML results confirm the tracking performance is getting worse when we use a larger NMS kernel. Meanwhile, the significant declines of FPs and increasing FNs indicate the accuracy of retrieving object locations is negatively correlated to the NMS kernel size. This is because only one response point with local maxima will be retrieved after a local NMS. For our response map, when objects are crowded or occluded with each other which is highly regular on MOT scenes, multiple distributions with max central response value will be clustered within a local image patch on response map. In our experiments, when multiple Gaussian-like distributions are clustered, we reserve the center point response of each distribution by taking the element-wise maximum value of overlap regions for normalization. Therefore, if NMS kernel size is larger than overlapped region, only one response point will be decoded, and the rest of foreground objects are filtered out. As shown in Fig. 8 (a), we obtained the best tracking performance when NMS kernel size was set to 3×3.

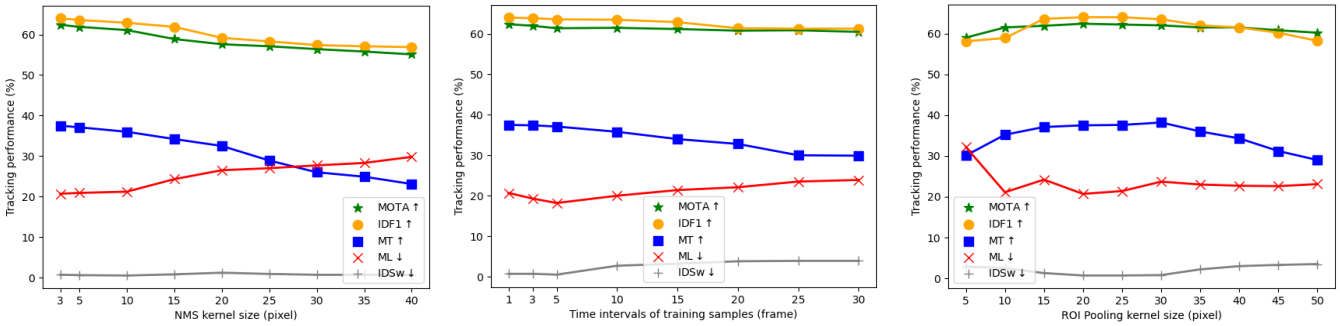

Fig. 8. Analyses on different experimental settings, best viewed in color. (a): Tracking performance vs. kernel sizes for local NMS, the best result we obtained was 3×3. (b): Tracking performance vs. time intervals for generating training samples, the best performance we obtained was using adjacent frames for training. (c): Tracking performance vs. kernel sizes for ROI Pooling, and the best result we obtained was 20×20.

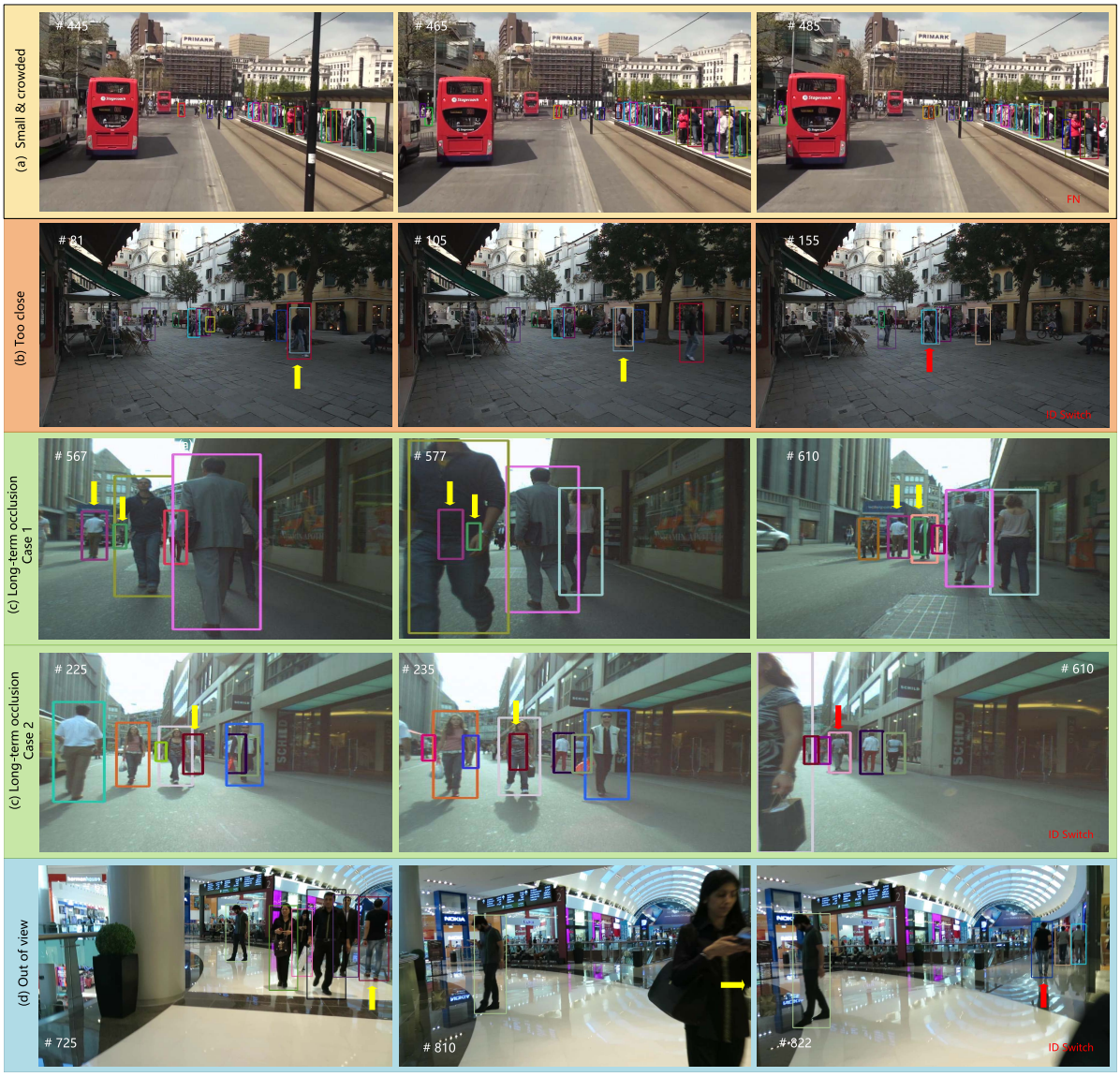

Fig. 9. Tracking results visualizations at challenging situations including failure cases, best viewed in color. From top to down: four different challenging situations including (a) small and very crowded objects filmed by a moving camera; (b) multiple times of occlusions with very close bbox centers; (c) long-term occlusions; (d) out of view. The yellow arrows point to the targets being investigated at each case, and the red ones point to the failure cases of ID Switches.

4) Experimental Settings for Motion Regressor: Our motion displacement regression sub-network using a frame-wise optical-flow-like offset to estimate the motion dynamics. We first compare the temporal intervals of optical flows used

for training motion regression sub-network to analyze the aspect of data diversity on tracking performance. As shown in table VII, the baseline method uses adjacent-frame optical flows to generate training samples. Methods interval 5, interval 10 and interval 20 denote the temporal intervals of extracted optical flow are $5, 10, 20$ frames respectively. The objective of involving larger intervals of training samples is to bring more intense motion variations for our motion model. The evaluation results on MOT dataset indicates that such experimental settings did not bring improvement to the tracking performance. The increasing ID Switches reveal the accuracy of our motion regressor is getting worse when we use larger time intervals of training samples. The errors of motion regression results may lead to mis-matches, false track initializations and fragments, which can explain the raising numbers of FNs. This is also related to the datasets used in experiments. The video sequences of MOT dataset [46] used in our experiment has a high frame rate of $15-30$ FPS, which means frame-wise motion displacements are rather small. We conducted the comparative analysis on MOT benchmark as shown in Fig. 8 (b), it is observed that using adjacent frame is sufficient enough for data diversity.

After sampling all displacements from frame-wise optical flow, our network resizes these patch-wise displacements into normalized ROIs through a ROI pooling layer, then regresses one accurate displacement value for each ROI. We further compare different scales of ROI to investigate the impact on motion regression accuracy. The ROI scale of baseline method in table VII is 20×20. From the comparative analysis in Fig. 8(c) we can observe that the medium size of ROI (20×20 to 30×30) brings us a better motion regression accuracy and tracking performance with low ID Switches.

5) Analyze on Post-Processing: We further compare the pose-processing strategy using target-dependent matching (baseline) approach with a target-independent one (without the Hungarian method). As described in section III$-C$, our target-independent matching approach replaces the Hungarian algorithm [45] by greedy matching using center point distance, and remove the matching cascade process. The bottom row in table VII shows the tracking performance of such ablation study. Without computing the IOU distance of two bounding boxes, our tracker still maintains a competitive tracking performance with 59.9 MOTA. This indicates the response locations learned from our response map are rather accurate, which can be confirmed by the slightly better MT and ML along with the almost unchanged FN comparing to the baseline method. The raising number of ID Switches is mainly because our tracker intends to handle the short-term occlusion. As for the long-term occlusion, e.g. a person is unseen from the scene for a long time and observed later, our tracker will not maintain the identity and thus a ID Switch is generated.

E. Discussion

We visualized the tracking results of our method at challenging situations as demonstrated in Fig. 9. Fig. 9 (a) illustrates a challenging situation when objects are very crowded with rather small scales, in which case our object locator is hard to retrieve all the positive signals and False Negatives are inevitable generated. Even so, the evaluation metric MOTA of our proposed tracker for this video (MOT16-14) is 38.5, which is still higher than the SOTA method tracktor++[63] whose corresponding MOTA is 32.5. As shown in Fig. 9 (b), when objects are too close, the output distributions of these positives on our global response map are largely overlapped, which may cause a FN after local NMS or a ID Sw. raised by inaccurate motion estimations. To further mitigate the suffer from this problem, more discriminative embeddings such as orientation could be introduced to our framework. Due to the video-based design with logical inference methodology, our proposed method is capable of estimating targets states during long-term occlusions as shown in Fig. 9 (c). However, targets being occluded are spatially invisible at the scene, and our network can only estimate the motion displacements upon history states, which may cause ID Sw. when targets have ambiguous moving patterns during occlusions as illustrated in Fig. 9 (c)-case 2. This can be further improved by introducing deep Re-ID features for affinity computations between image patches, at the cost of extra computational complexity. Our tracker only maintains track validity for unmatched targets with a maximum age A_{max}, when objects are out of view for more than A_{max} frames, our tracker will initialize a new track for them and results in ID Sw. when they reappeared in the scene (Fig. 9 (d)). Increasing A_{max} can improve the performance at this case, but may also incorrectly link objects to dead targets. We set A_{max} to 10 in our experiments out of time-efficiency and the evaluation results on MOT benchmarks are promising as well.

V. Conclusion

In this work, we introduce a novel object representation schema and a new network model to support end-to-end on-line MOT. The proposed approach is significantly different from existing tracking-by-detection and association based methods where the object detection step is only implicitly conducted by a more efficient object locating network, while other attributes of a tracked object, such as x/y and $\Delta x/\Delta y$, can be extracted by the corresponding sub-networks (the object locating sub-network and the motion displacement regression sub-network respectively) in one feed-forward propagation. The complete model is capable of describing the dynamic motions of multiple objects and can handle the entering/exiting/occlusion of objects robustly. The network generates a global response map as the intermediate output from which the trajectory of each object can be obtained. The proposed method is fast and accurate, and our evaluation based on the MOT benchmark show that the proposed tracker significantly outperforms many other state-of-the-art methods. We believe that such simple and effective algorithm can provide a new inspiration for the MOT community. Our extension to extract other attributes in a more complete motion tracking state space, such as width w and height h, as well as other attributes including $orientation$, $depth$, etc., is on-going.

References

[1] X. Wan, J. Wang, Z. Kong, Q. Zhao, and S. Deng, "Multi-object tracking using online metric learning with long short-term memory," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 788–792.

[2] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 749–765.

[3] G. Ding, W. Chen, S. Zhao, J. Han, and Q. Liu, "Real-time scalable visual tracking via quadrangle kernelized correlation filters," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 140–150, Jan. 2018.

[4] B. Zhang *et al.*, "Latent constrained correlation filter," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1038–1048, Mar. 2018.

[5] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[6] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. CVPR*, Jun. 2011, pp. 1201–1208.

[7] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5620–5629.

[8] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 100–111.

[9] S. Zhang, J. Wang, Z. Wang, Y. Gong, and Y. Liu, "Multi-target tracking by learning local-to-global trajectory models," *Pattern Recognit.*, vol. 48, no. 2, pp. 580–590, 2015.

[10] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," 2015, *arXiv:1501.04587*. [Online]. Available: https://arxiv.org/abs/1501.04587

[11] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[12] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

[13] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1918–1925.

[14] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 84–99.

[15] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[16] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[17] P. Scovanner and M. F. Tappen, "Learning pedestrian dynamics from the real world," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 381–388.

[18] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–311.

[19] X. Wan, J. Wang, and S. Zhou, "An online and flexible multi-object tracking framework using long short-term memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 1230–1238.

[20] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear LSTM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 200–215.

[21] J. Zhang *et al.*, "Multiple object tracking by flowing and fusing," 2020, *arXiv:2001.11180*. [Online]. Available: https://arxiv.org/abs/2001.11180

[22] J. Zhang, S. Zhou, J. Wang, and D. Huang, "Frame-wise motion and appearance for real-time multiple object tracking," 2019, *arXiv:1905.02292*. [Online]. Available: https://arxiv.org/abs/1905.02292

[23] Y. Xu, A. Sep, Y. Ban, R. Horaud, L. Leal-Taixe, and X. Alameda-Pineda, "How to train your deep multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6787–6796.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[25] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.

[26] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

[27] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. OE-8, no. 3, pp. 173–184, Jul. 1983.

[28] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[29] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, Nov. 2015.

[30] A. Dehghan, Y. Tian, P. H. S. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1146–1154.

[31] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2034–2041.

[32] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1926–1933.

[33] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3682–3689.

[34] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.

[35] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 33–40.

[36] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 549–565.

[37] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4225–4232.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[39] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 734–750.

[40] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[41] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3203–3212.

[42] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2462–2470.

[43] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2016, pp. 779–788.

[45] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[46] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: https://arxiv.org/abs/1603.00831

[47] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2009.

[48] 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

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[50] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2012, pp. 611–625.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[52] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2497–2504.
[53] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 21–37.
[54] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 3988–3998.
[55] P. Chu, H. Fan, C. C. Tan, and H. Ling, "Online multi-object tracking with instance-aware tracker and dynamic model refreshment," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 161–170.
[56] H. Shen, L. Huang, C. Huang, and W. Xu, "Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking," 2018, *arXiv:1808.01562*. [Online]. Available: https://arxiv.org/abs/1808.01562
[57] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 482–490.
[58] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang, "Multi-object tracking with multiple cues and switcher-aware classification," 2019, *arXiv:1901.06129*. [Online]. Available: https://arxiv.org/abs/1901.06129
[59] L. Ma, S. Tang, M. J. Black, and L. Van Gool, "Customized multi-person tracker," in *Proc. Asian Conf. Comput. Vis.* New York, NY, USA: Springer, 2018, pp. 612–628.
[60] L. Chen, H. Ai, R. Chen, and Z. Zhuang, "Aggregate tracklet appearance features for multi-object tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1613–1617, Nov. 2019.
[61] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.
[62] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2018.
[63] R. Henschel, L. Leal-Taixe, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1428–1437.
[64] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3269–3280, Nov. 2018.
[65] P. Chu and H. Ling, "FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6172–6181.
[66] R. Henschel, Y. Zou, and B. Rosenhahn, "Multiple people tracking using body and joint detections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–10.
[67] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.
[68] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 17–35.
[69] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2953–2960.
[70] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.
[71] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 36–42.
[72] N. Mahmoudi, S. M. Ahadi, and M. Rahmati, "Multi-target tracking using CNN-based features: CNNMTT," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7077–7096, Mar. 2019.
[73] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1809–1814.

Xingyu Wan received the B.S. degree in physics from Xiamen University, China, in 2015. He is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China. His research interests include deep learning, pattern recognition, and computer vision, with a focus on image super-resolution, video analyzing, object detection, and visual tracking.

Jiakai Cao was born in Jiangsu, China, in 1999. He is currently pursuing the B.S. degree in computer science with Xi'an Jiaotong University, Xi'an, China. His research interests include deep learning, computer vision, and multi-object tracking.

Sanping Zhou (Member, IEEE) received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2020. From 2018 to 2019, he was a Visiting Ph.D. Student with the Robotics Institute, Carnegie Mellon University. He is currently an Assistant Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include machine learning, deep learning and computer vision, with a focus on medical image segmentation, person re-identification, salient object detection, image classification, and visual tracking.

Jinjun Wang received the B.E. and M.E. degrees from Huazhong University of Science and Technology, China, in 2000 and 2003, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. From 2006 to 2009, he was with NEC Laboratories America, Inc., as a Research Scientist, and from 2010 to 2013, he was with Epson Research and Development, Inc., as a Senior Research Scientist. He is currently a Professor with Xi'an Jiaotong University. His research interests include pattern classification, image/video enhancement and editing, content-based image/video annotation and retrieval, and semantic event detection.

Nanning Zheng graduated from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975. He received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He jointed Xi'an Jiaotong University in 1975, where he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems. He became a member of Chinese Academy of Engineering in 1999. He is also Chinese Representative on the Governing Board of the International Association for Pattern Recognition.