# Spatio-Temporal Point Process for Multiple Object Tracking

Tao Wang, Kean Chen, Weiyao Lin, *Senior Member, IEEE*, John See, *Senior Member, IEEE*, Zenghui Zhang, *Senior Member, IEEE*, Qian Xu, and Xia Jia

***Abstract*—Multiple object tracking (MOT) focuses on modeling the relationship of detected objects among consecutive frames and merge them into different trajectories. MOT remains a challenging task as noisy and confusing detection results often hinder the final performance. Furthermore, most existing research are focusing on improving detection algorithms and association strategies. As such, we propose a novel framework that can effectively predict and mask-out the noisy and confusing detection results before associating the objects into trajectories. In particular, we formulate such "bad" detection results as a sequence of events and adopt the *spatio-temporal point process* to model such events. Traditionally, the occurrence rate in a point process is characterized by an explicitly defined intensity function, which depends on the prior knowledge of some specific tasks. Thus, designing a proper model is expensive and time-consuming, with also limited ability to generalize well. To tackle this problem, we adopt the convolutional recurrent neural network (conv-RNN) to instantiate the point process, where its intensity function is automatically modeled by the training data. Furthermore, we show that our method captures both temporal and spatial evolution, which is essential in modeling events for MOT. Experimental results demonstrate notable improvements in addressing noisy and confusing detection results in MOT data sets. An improved state-of-the-art performance is achieved by incorporating our baseline MOT algorithm with the spatio-temporal point process model.**

***Index Terms*—Multiple object tracking, recurrent neural networks, spatio-temporal point processes.**

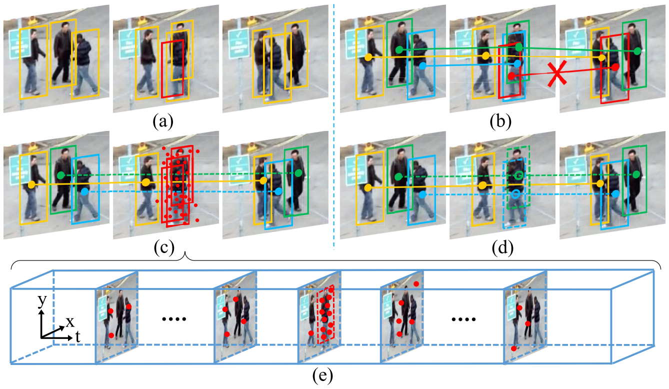


Fig. 1. Example of a noisy object detection. (a) Detection results, where a potential noisy detection (red box) occurs in the scene. (b) As a result, the red trajectory is unnecessarily tracked. This is an incorrect result. (c) Our approach can predict the area which is likely to contain the noisy detections and avoid them in the association process. (d) Tracking result using our method, where the dashed line boxes are generated by linear interpolation. (e) Prediction is generated by the proposed point process model.

## I. Introduction

MULTIPLE object tracking (MOT) is one of the fundamental problems in computer vision, which is important in many applications like intelligent video surveillance, behavior analysis, automatic driving, and robotics. MOT constitutes the task of modeling the relationship of detected objects among consecutive frames and then merging them into different trajectories [1]–[3]. This task remains challenging, one major issue is that some "bad" detection results always hinder the performance of MOT. In general, such "bad" results can be divided into two different types: 1) noisy detection results, i.e. false positives in object detection, and 2) confusing detection results, i.e. two highly overlapping objects with similar appearances. Two such examples are shown in Figs. 1 and 2. It can be seen that both the noisy and confusing objects had misguided the matching process. Furthermore, we observe that most of the failure examples in MOT are caused by them, directly or indirectly. Hence, the main problem in MOT that needs to be addressed can realistically be reduced to: *how to effectively handle these "bad" detection results?*

Some existing methods in MOT such as [1] and [3]–[10] improved the tracking performance by introducing more robust object association strategies. More specifically, some advanced techniques for cost function and corresponding optimization algorithm have been developed to associate objects in different frames. However, these methods do not explicitly model the "bad" detection results, so they can be confused by the objects with high similarity in appearance and motion. Other methods including [11]–[16] focused their attention on different feature representations and metrics for detected objects. These methods have better accuracy in normal scenes but are still affected by the noisy detection results. Some works

Manuscript received October 15, 2019; revised March 1, 2020 and April 3, 2020; accepted May 14, 2020. This work was supported in part by the China Major Project for New Generation of AI under Grant 2018AAA0100400, in part by the National Natural Science Foundation of China under Grant 61971277, in part by ZTE Industry-Academia-Research Cooperation Funds and State Key Laboratory of Mobile Network and Mobile Multimedia Technology, and in part by CREST Malaysia under Grant T03C1-17. *(Corresponding author: Weiyao Lin.)*

Tao Wang, Kean Chen, Weiyao Lin, and Zenghui Zhang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wang_tao1111@sjtu.edu.cn; ckadashuaige@sjtu.edu.cn; wylin@sjtu.edu.cn; zenghui.zhang@sjtu.edu.cn).

John See is with the Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia (e-mail: johnsee@mmu.edu.my).

Qian Xu and Xia Jia are with ZTE Corporation, Shenzhen 518057, China (e-mail: xu.qian5@zte.com.cn; jia.xia@zte.com.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2997006

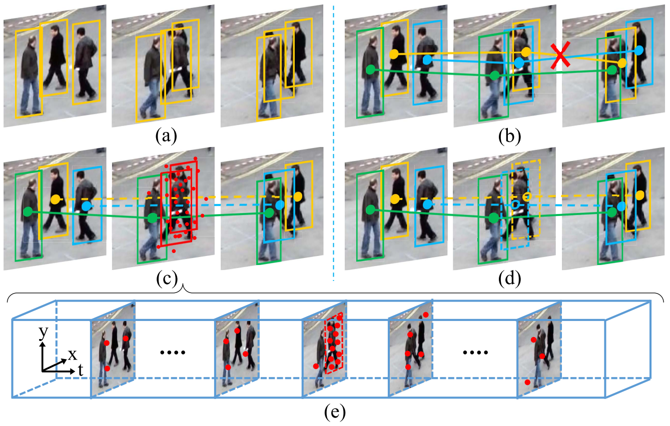


Fig. 2. Example of a confusing object detection. (a) Detection results. (b) Incorrect tracking result due to objects that are overlapping in close proximity (yellow box and blue box). (c) Our approach can predict the area which is likely to contain the confusing objects and avoid them in the association process. (d) Tracking result by our method, where the dashed line boxes are generated by linear interpolation. (e) Prediction is generated by the proposed point process model.

like [17]–[22] adopted more accurate object detectors in attempt to reduce these noisy detections, while the performances are still hindered by confusing detections such as the highly overlapping objects.

To tackle this issue in MOT, we propose a framework that can effectively predict and mask-out these "bad" detection results before associating the objects into trajectories. First, we note that the "bad" detections can be formulated as a sequence of events that happen in different frames and locations. Thus, we need to infer when and where these events are likely to happen, given the feature of current frame and the historical behavior of the detector as prior information. More specifically, we model these events based on the pixels that are inside the bounding boxes of "bad" detections. Such events are distributed across the spatio-temporal tube of a video and are generated based on complicated latent mechanisms [23], [24], which is hard to capture through simple modeling. Because these events happen in the motion of objects, so there exists relationship between these events which actually reflects the motion information of the objects. In other words, we can detect these bad detections more accurately and improve the performance of MOT by obtaining the relationship between these events. Moreover, we make the following observations.

1) Noisy object detections are more likely to appear in the area where there are already some noisy detections in the previous frames.
2) If there are confusing detections among people in a group who walk closely or dress similar to others, confusing detections are more likely to appear in these people in the subsequent frames.

In this article, we introduce the use of *spatio-temporal point process* to deal with such events.

Point process [25] is a powerful tool for modeling the real-world sequential data, which has lots of applications in many fields, such as finance [26], equipment maintenance [27], [28], and social network [29], [30]. A point process is characterized by its conditional intensity function, which presents the occurrence rates of some class of events conditioned on the historical events. Traditionally, the intensity function can be explicitly defined based on prior knowledge of event data and latent mechanisms of the process. The intensity function usually consists of two parts [31]: an exogenous intensity that describes factors driven by the inherent and often time-varying occurrence rate for a type of events; an endogenous intensity which describes the triggering effect from the previous events. This parametric strategy has been widely adopted in many classic models, such as Poisson processes [32], Hawkes processes [33], and self-correcting processes [34]. However, there are three issues that need to be handled in the case of MOT.

1) Designing a proper parametric model is expensive and time-consuming, since it requires expert domain knowledge and experience if it were to be modeled manually. Besides, the generalization ability is also limited.
2) It is difficult to properly capture the dynamics of influence from historical information given the complicated nature of data patterns in MOT task. On the other hand, it is also hard for traditional point processes to incorporate other heterogeneous data, such as time series [31] associated with event sequences.
3) Furthermore, the detection events in MOT occur in different frames and different locations in the scene, which requires capturing both temporal and spatial evolution.

To tackle these problems, we adopt the *spatio-temporal point process* to model the detection events, where a convolutional recurrent neural network (conv-RNN) is proposed to instantiate the point process. The main characteristics of our method are shown below and in Fig. 3.

1) In our method, the intensity function is automatically modeled by the training data, without requiring expert knowledge and experience.
2) We propose a two-stream RNN framework to handle two different inputs, i.e., time series and event sequence, with a novel time-evolving mechanism to align and merge these two input data. This enables our model to capture the complex dynamics of influence from historical information.
3) We incorporate the use of convolution operation in RNN, which enables spatial diffusion for the historical information of events.

Coupled with the capacity of RNN in modeling temporal dependence, the proposed conv-RNN-based point process is equipped with the capability of capturing both temporal and spatial evolution. The main contributions of this article are summarized as follows.

1) We propose a novel framework in MOT that can effectively predict and mask-out the noisy and confusing detection results before associating the objects into trajectories. The "bad" detection results are formulated as a sequence of events and are modeled by the *spatio-temporal point process*.
2) We introduce a two-stream pipeline to handle two synchronous and asynchronous inputs (i.e., time series and event sequence) with a novel-evolving mechanism for merging.

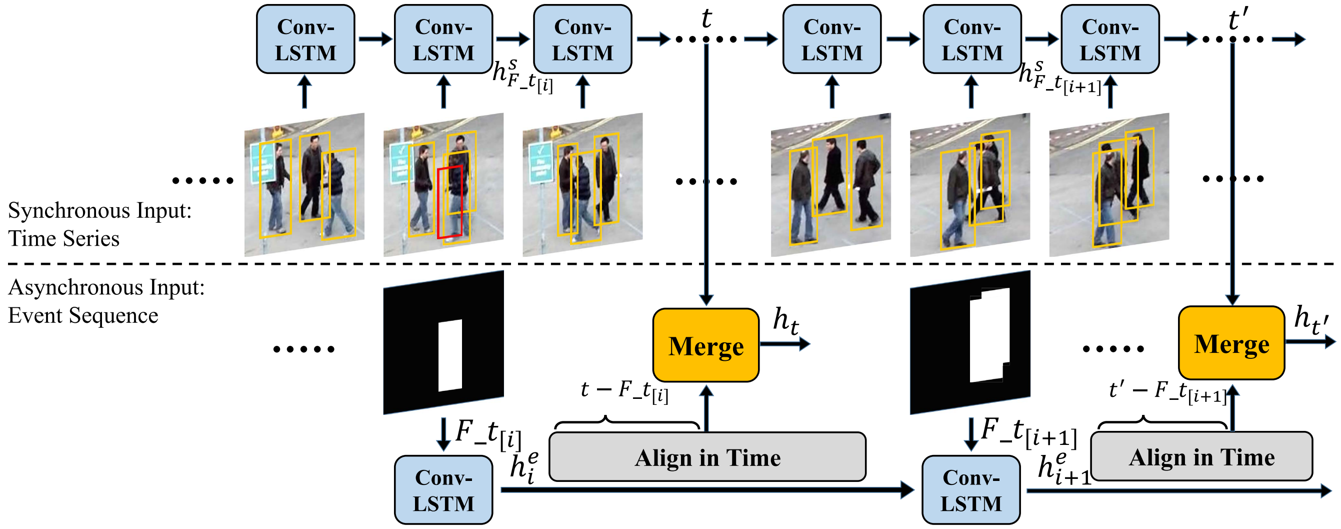


Fig. 3. Framework of the proposed point process model. There are two RNNs in our model. The first one is for modeling the time series, which represents the background information and features at each time. In our approach, the features and background information are the raw image and the detection results in this frame. The second one is for modeling the event sequence, which represents the historical events and their time stamps. In our approach, the events are defined to be the pixels which are inside the "bad" detection results. The feature $h_i^e$ which encodes the event sequence is aligned in time and synchronized with the feature $h_t^s$ from time series. $[i]$ denotes the $i$th equivalence class of events that have the same time stamp so that $t_{[i]}$ means the $i$th frame that contains at least one event.

3) We propose a convolutional-based RNN to instantiate the point process, where the temporal and spatial evolution are well captured.
4) We show notable performance improvement by incorporating the proposed method into state-of-the-art MOT algorithms across different metrics and benchmark data sets.

## II. Related Works

### A. Multiple Object Tracking

MOT is widely applied in many fields like intelligent video surveillance, behavior analysis, automatic driving, and robotics [12], [35]. MOT involves the process of modeling the relationship of detected objects among consecutive frames and then merging them into different trajectories [1]–[3]. This task remains challenging due to several problems. An obvious observation is that certain "bad" detection results remain the bottleneck for better performance in MOT. Generally, these "bad" results can be divided into two different types, reflecting two main causes: 1) *noisy* detection results, i.e., false positives in object detection, and 2) *confusing* detection results, i.e., two highly overlapping objects with similar appearances. Among existing works, there are three known aspects of how these issues have been handled.

1) Some existing methods in MOT such as [1] and [3]–[9] improve the tracking performance by introducing more robust object association strategies. For example, Berclaz *et al.* [4] proposed a constrained flow optimization problem to model the objects association, and use k-shortest path algorithm [36] to solve this problem. Yang *et al.* [3] modeled the data association into a min-cost multicommodity network flow which fuses both global optimization and local optimization.
2) Other methods such as [11]–[15] focus their attention toward studying different feature representations and metrics in attempt to mitigate these errors. In addition, Gao and Jiang [37] proposed an attention-based appearance model to obtain a better similarity metric.
3) Some research works [17]–[22] adopted more accurate object detectors to reduce these problems, particularly noisy detections. For instance, Henschel *et al.* [17] proposed a multidetector to track pedestrian through fusing the detection of head and body. Yu *et al.* [18] proposed a high-performance detector for MOT which combines skip pooling [38] and multiregion strategies [39].

Although these methods give valuable results in MOT, their performances are still limited due to intrinsic limitations; in particular, these methods contained unaddressed deficiencies. First, these methods do not explicitly model "bad" detection results. As such, incorrect tracking caused by confusion of objects with high similarity in appearance and motion cannot be learned or modeled. Second, these methods demonstrated good tracking performance in normal scenes whereas in scenes with noisy detection results, they remain highly susceptible to erroneous tracking. However, the proposed approach explicitly models the dynamic and relationship of the "bad" detections and is able to predict and mask-out them before associating the objects into trajectories.

### B. Spatio-Temporal Point Processes

Temporal point process has been widely applied in many fields such as finance [26], equipment maintenance [27], [28], and social network [29], [30]. The point process is characterized by the conditional intensity function, which presents the occurrence rates of events conditioned on the historical data. Traditionally, the intensity function is explicitly defined based on prior or expert knowledge about event data and latent mechanisms of the process. However, in many tasks, it is hard to design a parametric model by hand. Recently, several research works [31], [40] studied the use of a neural network-based model for the temporal point process. Du *et al.* [40] proposed a semiparametric model for the point process where the events are treated as a univariate point process and the event types are treated as marks that are associated with events. Their key idea is to view the intensity function of a temporal point process as a nonlinear function of the event history and use an RNN to automatically learn a representation of influences from the event history. Xiao *et al.* [31] introduced recurrent point process networks which instantiate temporal point process models with temporal RNNs. Their key idea is to model the intensity function by two RNNs: one temporal RNN captures the relationships among events, while the other RNN updates the intensity functions based on time series. On the other hand, the spatio-temporal point process [41] reveals two main pieces of information (space and time) about the data considered. The data are treated as the realization of a random collection of points, which evolves in space and time. It can be viewed as a temporal point process with further (spatial) dimensions.

Spatio-temporal point process also has wide applications such as earthquakes [42] and disease outbreaks [43]. For the case of the MOT task, this is potentially feasible since we are also interested in both the spatial and temporal evolution of detected objects. In contrast to the aforementioned existing schemes which define a neural network model for the temporal point process, we propose conv-RNN to instantiate the point process. The convolution operation in RNN enables spatial diffusion for the historical information of events. Combined with the capacity of RNN in modeling temporal dependencies, our conv-RNN-based point process can effectively capture both temporal and spatial evolution.

### C. Recurrent Neural Networks

RNN [44]–[46] is a kind of neural network tailored for modeling sequences such as time-series data. RNN allows connections among hidden units to be associated with a time delay, which is useful to encode historical information and capture the relationship between events that evolve over time. However, a vanilla RNN model is difficult to train and does not handle long-range dynamics well due to the *vanishing gradient* and *exploding gradient* problems [47]. To this end, the long short-term memory (LSTM) is proposed, which has shown to be stable and powerful for modeling long-range dependencies. The key idea of LSTM lies in the memory cell $c_t$, which acts as an accumulator of the state information. It allows gradient to be trapped in the cell and be prevented from vanishing too quickly [48]. Traditional LSTM has 1-D vectors for the cell input-output and hidden state. Such LSTMs contain too much redundancy for spatial data. To address this problem, Xingjian *et al.* [48] proposed conv-LSTM which has convolutional structures in both the input-to-state and state-to-state transitions. Furthermore, by stacking multiple conv-LSTM layers, we can build a network model that is feasible for spatio-temporal sequence forecasting problems such as the MOT task. In this article, we design a variant of the described conv-RNN model, which can handle different inputs for both long- and short-term information, and can incorporate the spatio-temporal varying components to build the intensity. The proposed method will be elaborated in detail in Section IV after describing the MOT task in Section III.

## III. Preliminaries

### A. Problem Definition

The input data are a sequence $C = \{I_1, I_2, \dots\}$ of consecutive frames in a video, combined with the object detection results $D_j = \{d_1^j, d_2^j, \dots\}$ in each frame $I_j$. Note that the detection results $D_j$ are generated by some baseline detection algorithms and are not guaranteed to be entirely correct. For example, the false positive detections that always occur in crowded scenes are regarded as noisy detections. Besides, other erroneous detection results are caused by confusion from detected objects of similar appearances which overlapped with each other during motion. The MOT task aims to model the relationship among these detected objects and associate them into trajectories. Hence, these noisy or confusing detection results will hinder the performance of MOT. In this article, we adopt the spatio-temporal point process to address this issue.

Let $B_j = \{b_1^j, b_2^j, \dots\}$ denote the set of "bad" detections in each frame $I_j$ (note that $B_j \subseteq D_j$). In our spatio-temporal point process, we define the events $e$ to be the pixels $e = (t, x, y)$ that are inside some "bad" detection results $b_i^t$ in frame $I_t$. We use $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ to denote the set of events from all frames. In general, the dynamics of the spatio-temporal point process is captured by its conditional intensity function $\lambda^*(t, x, y)$ (where $*$ emphasizes that this function is conditional on the history), which can be defined as

$$\lambda^*\left(t', x', y'\right) = \frac{\partial^3 \; E[N(t, x, y)|\mathcal{H}_t]}{\partial t \partial x \partial y}|_{t=t', x=x', y=y'} \tag{1}$$

where $\mathcal{H}_t$ denotes the history of events before time $t$, and $N(t, x, y)$ is the counting function which counts the number of events with spatial coordinates less than $(x, y)$ and temporal coordinates less than $t$, while $E[N]$ is the expectation of the counting function. Since we assume that the point process satisfies the regularity condition

$$P\{N([t, t+\epsilon_1], [x, x+\epsilon_2], [y, y+\epsilon_3]) > 1\} = o(\epsilon_1\epsilon_2\epsilon_3) \tag{2}$$

where $P$ measures the probability of event (its argument), $[t, t+\epsilon_1], [x, x+\epsilon_2], [y, y+\epsilon_3]$ indicates a small area around the point $(t, x, y)$ while $o(\epsilon_1\epsilon_2\epsilon_3)$ refers to a term of higher-order infinitesimal w.r.t. $\epsilon_1\epsilon_2\epsilon_3$. Equation (2) shows that in the point process there will be at most one event within a short time and in a small area. The intensity function $\lambda^*(t, x, y)$ can also be interpreted as the conditional probability density that an event occurs at $(t, x, y)$. Thus, the probability density for an event sequence $\{e_1, e_2, \dots\}$ can be written as

$$f(e_1, e_2, \dots) = \prod_j f^*\left(t_j, x_j, y_j|\mathcal{H}_{t_j}\right) \tag{3}$$

where

$$f^*\left(t, x, y|\mathcal{H}_{t_j}\right) = \frac{\lambda^*(t, x, y)}{\exp\left(\int_{\tilde{t}_j}^{t} \int_{u,v} \lambda^*(\tau, u, v) d\tau du dv\right)}. \tag{4}$$

Here, $\widetilde{t_j}$ denotes the maximal time stamp $t_k$ that satisfies $t_k < t_j$, the denominator presents the probability that no new event occurs up to time $t_j$ since $\widetilde{t_j}$. Point processes are always learned by maximizing the likelihood function; whereby its implementation is not specified. Thus, one problem is how to define and implement the intensity function $\lambda^*(t, x, y)$ in the context of MOT.

### *B. Tracklet-Based Algorithm*

In this article, our baseline tracking method follows the tracklet-based strategy [11], [22], [24] to associate objects from each frame into trajectories. Similar to [24], the baseline tracking method contains three steps. The first step is *tracklet construction*, which aims to merge highly related objects into high-confidence tracklets. Note that we take a strict threshold of appearance similarity and motion similarity for object matching. This ensures that the generated tracklets are highly reliable and consistent. Then, these tracklets are treated as basic units for the subsequent clustering step. The second step is *tracklet similarity calculation*. In some works [1], [3], tracklet similarity is calculated directly from features of the terminal object (i.e., object on the last frame of the tracklet), which cannot capture information of the entire tracklet. Instead of using only the terminal object, we determine tracklet similarity by using all the appearance information in the tracklets. More details can be found in [24]. The third step is *tracklet clustering*. After obtaining the tracklets and corresponding similarities, the clustering process associates these tracklets into complete trajectories. The basic idea in [24] is that cluster centers are characterized by a higher local density than their neighbors and by a relatively larger distance from other sample points with higher local densities. The local density of $i$th tracklet is defined as

$$\rho_i = \sum_{j:O(i,j)=0} H\left(s_{ij} - s_c\right) \tag{5}$$

where $s_c$ is a threshold which is always set to 0.5, $H(x)$ is the Heaviside step function, $s_{ij}$ denotes the similarities between $i$th and $j$th tracklets. The constraint $O(i, j) = 0$ ensures that these two tracklets have no overlap in the temporal domain. The maximum similarity of $i$th tracklet is defined as

$$\delta_i = \max_{j:\rho_j > \rho_i, O(i,j)=0} s_{ij}. \tag{6}$$

Then, the tracklets that have a maximum similarity $\delta_i < s_c$ are marked as cluster centers such that the similarities between any two cluster centers are always lower than $s_c$. Finally, the remaining tracklets are assigned to the same cluster according to the most similar tracklet of higher density.

### *C. Combination of Tracking and Point Process*

The whole proposed tracking framework is shown in Fig. 4. At first, we use the spatio-temporal point process to model intensity function $\lambda^*(t, x, y)$ for each frame $I_j$ of the sequence $C$. Then, we predict "bad" detection events for each frame and remove those detection bounding boxes which contain events more than a given threshold. After this step, we get better detection results $D_j^*$ for each frame of the sequence $C$. Then, $D_j^*$ will be used as the input of our tracking baseline to construct short tracklets. Finally, the tracklets are clustered and associated with generating complete trajectories.

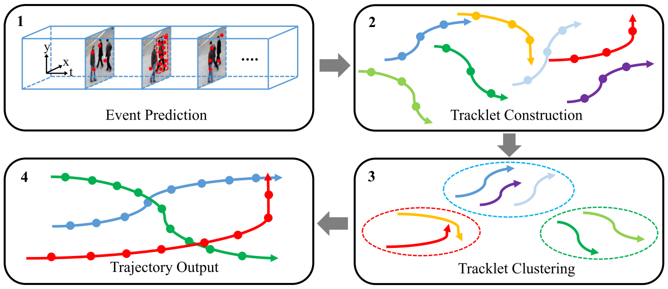


Fig. 4. Framework of the tracking method. First, the spatio-temporal point process is adopted to predict "bad" detection events. The "bad" detection boxes, which cover too many events, will be directly deleted from each frame. Then, the detections (dots) are linked to form different tracklets (arrows). Finally, the tracklets are clustered and associated with generating complete trajectories.

It should be noted that although an event pixel can be contained by more than a single box, the definition of events in our method still matches well with that in the point process. In our method, "bad" events are not "bad" detection boxes but the "bad" pixels in detection boxes. So, if a "bad" event on a pixel is contained by many bounding boxes, it still represents *one* event, not many events. Furthermore, when an event happens in the overlapped area of two bounding boxes, it also means that this event decreases the confidence of both boxes.

## IV. Method

In this section, we describe in detail the method which characterizes the proposed neural network model for the spatio-temporal point process.

As shown in Fig. 3, we have two RNNs in the proposed model. The first one, a synchronous network, is for modeling the time series [31], which represents the background information and features at each time step. We use the raw image and the detection results in each frame to model the features and background information. The second one, an asynchronous network, is for modeling the event sequence, which represents the historical events and their time stamps. "Bad" detections are defined in two ways: (1) noisy detections, which are bounding boxes that are false positives or inaccurate detections, (2) confusing detections, which are bounding boxes that are easily mistaken for other objects. The labeling strategy for the "bad" detection is shown in Section V. The events are defined to be the pixels which are inside the "bad" detection results. We take the binary map as the input to each recurrent unit (LSTM), where the pixels are masked to one if the event occurs. The feature $h_i^e$ which encodes the event sequence is aligned and synchronized with the feature $h_t^s$ from time series. Then, we merge the features from two RNNs and calculate the intensity function based on it. After obtaining the intensity function, we can infer the event through sampling on the probability density function [cf., (4)]. In the testing phase, for each bounding box $b$, we calculate the ratio $r_b$ between the number of events in it and its area. Then, a detection $b$ is treated as a "bad" detection if $r_b$ is larger than some threshold.

### A. Synchronous RNN

The synchronous RNN takes in time series as input data. We have two kinds of time series: one is the raw images; one is the binary maps for detection results, where the pixels that are inside a detected bounding box are masked to 1, otherwise 0. As shown in [49], though the mask of the bounding box is a weaker label than the segmentation mask of the object, it still provides sufficient information to train the segmentation task. However, we intend for the proposed model to focus on learning potential areas of "bad" detections instead of actual segmentation of objects. Thus, the binary maps of bounding boxes are used as auxiliary input data instead of a training label.

At each step $t$, we first send the raw image $I_t$ and the binary map $M_t^D$ of detection results to a CNN (e.g., VGG, ResNet) to extract feature maps. Then, the feature maps are merged and fed to the recurrent unit (conv-LSTM) to extract temporal features. Therefore, the hidden states $h_t^s$ for the time series can be obtained by

$$\left(h_t^s, c_t^s\right) = \mathrm{ConvLSTM}\left(h_{t-1}^s, c_{t-1}^s, \psi_1\left(I_t, M_t^D\right)\right) \tag{7}$$

where $\psi_1$ denotes the feature extractor CNN, $c_t^s$ denotes the cell status and both $h_t^s$ and $c_t^s$ are feature maps. Note that such input data are sampled uniformly over time, which can reflect the exogenous intensity in the point process.

### B. Asynchronous RNN

The asynchronous RNN aims to capture the relation between events. It takes in an event sequence as input data. In this spatio-temporal point process for MOT, each event provides both spatial and temporal features. For spatial feature, our strategy is similar to that in Section V-A. We use the binary map to represent the spatial information of events that occur in a frame, i.e., those pixels that are inside the "bad" detections are masked to 1, otherwise 0. For the temporal feature, we use the interevent duration $F\_t_{[i]} - F\_t_{[i-1]}$ for each event occurring at time $F\_t_{[i]}$. Here, $[i]$ denotes the $i$th equivalent class of events, so $F\_t_{[i]}$ denotes the $i$th frame that contains at least one event. We reformulate the scalar $F\_t_{[i]} - F\_t_{[i-1]}$ to a map that has the same size as the raw image and all pixels are represented by this scalar. Then, these two maps are concatenated and passed through a CNN feature extractor and conv-LSTM to obtain the hidden state as follows:

$$h_i^e = \mathrm{ConvLSTM}\left(h_{i-1}^e, c_{i-1}^e, \psi_2\left(M_i^B, F\_t_{[i]} - F\_t_{[i-1]}\right)\right) \tag{8}$$

where $\psi_2$ denotes the CNN feature extractor, $M_i^B$ denotes the binary map of "bad" detection results, $c_i^e$ denotes the cell status, and both $h_i^e$ and $c_i^e$ are feature maps. Note that the events are always sparsely located and unevenly distributed in the time domain, which shows that the sequences are asynchronous in nature and the temporal intervals can be very long. Thus, one advantage of such asynchronous RNN is that it can better capture long-term dependencies of events, which can reflect the endogenous intensity in the point process.

### C. Align and Merge

We aim to construct the intensity function based on the features from both time series and event sequence. To this end, the asynchronous features should be first aligned and synchronized with the time series. Traditionally, one may consider multiplying a decaying function $\gamma(t - F\_t_{[i]})$ with the feature $h_i^e$ to simulate the influence from historical events to current time, where $\gamma(t)$ can be specified by $\exp(-t)$. However, the latent dynamics of the point process in MOT are still unknown. Directly specifying the evolving function $\gamma(t)$ in such a manner may lead to the model misspecification problem [40]. Therefore, we use a three-layer multilayer perceptron (MLP) whose input is the concatenation of $t - F\_t_{[i]}$ and $h_i^e$ to learn the evolving function $\psi_3(\cdot, t)$, where the model capacity can be sufficiently large to cover any dynamic patterns. Assuming current time is $t$ and the latest hidden state of the event $e$ is $h_i^e$, we have

$$\hat{h}_t^e = \psi_3\left(h_i^e, t - F\_t_{[i]}\right) \tag{9}$$

where $\hat{h}_t^e$ denotes the aligned event feature at time $t$. After that, we merge the outputs from both time series and event sequence to obtain $h_t = [h_t^s, \hat{h}_t^e]$. Finally, the intensity function $\lambda(t)$ can be formulated as

$$\lambda^*(t) = \sigma\left(w^s h_t^s + w^e \hat{h}_t^e\right) \tag{10}$$

where $\lambda^*(t)$ in this equation is an intensity map, the $w^s h_t^s$ term reflects the exogenous intensity, the $w^e \hat{h}_t^e$ term reflects the endogenous intensity, and $\sigma$ denotes the activation function. A reasonable choice of the activation function would be one that ensures that the intensity function is non-negative and almost linear when the input is much greater than 0. Several activation functions fit this purpose: the *softplus* function, defined as $\log(\exp(x) + 1)$ or exponential linear unit *(elu)*, defined as $\mathrm{elu}(x) + 1$ where $\mathrm{elu}(x) = x$ when $x \geq 0$ and $\mathrm{elu}(x) = \exp(x) - 1$ when $x < 0$.

### D. Learning Method

Given the sequence of events $E = \{e_1, e_2, \ldots, e_n\}$, we obtain the intensity function $\lambda^*(t, x, y)$ by (10). Then, the log-likelihood function can be formulated based on (3) and (4) as follows:

$$\log f = \sum_j \log \lambda^*\left(t_j, x_j, y_j\right) - \int_{t_0}^{t_n} \int_{u,v} \lambda^*(\tau, u, v) d\tau du dv. \tag{11}$$

The first part is the accumulative summation of log-intensity function for those events occurring at $\{(t_1, x_1, y_1), (t_2, x_2, y_2), \ldots, (t_n, x_n, y_n)\}$. The second part is the integral of intensity over space and time where no event happens. In our implementation, we set the spatio-temporal resolution to be one and adopt a discrete approximation of (11)

$$\log f = \sum_j \log \lambda^*\left(t_j, x_j, y_j\right) - \sum_{(\tau,u,v) \neq \left(t_j, x_j, y_j\right)} \lambda^*(\tau, u, v). \tag{12}$$

This objective function is differentiable and can be efficiently optimized by stochastic gradient descent.

### *E. Testing Method*

In the testing phase, for each video, we put the detection results of each frame and output of the previous frames into the proposed point process model. Through this way, we obtain an intensity map for each frame $t$ which is then used to infer the event by sampling on the probability density function. For each bounding box $b_i$ in frame $t$, we count the number of events in it and calculate the ratio $r_i$ between event number and box area. If $r_i$ is larger than a given threshold we will consider $b_i$ as a "bad" detection and remove it from frame $t$. After repeating the above process for each frame of the testing video, all "bad" detections would have been removed and we proceed to use the remaining detection results as the input of the tracking method to form tracklets.

## V. Experiments

### *A. Experimental Settings*

*1) Data Sets:* Our experiments are performed on two benchmark data sets: MOT16 [50] and MOT17. MOT16 data set contains seven training and seven testing sequences. MOT17 has the same video sequences as MOT16, with three different detection sets (DPM [51], faster-RCNN [52], and SDP [53]). The video sequences are captured by both static and moving cameras, with different scenes and resolutions. The viewpoint can also vary significantly from each other, e.g., the camera may be overlooking the scene from a high position, a medium position (at pedestrian's height), or at a low position (at ground level). Also, various forms of object occlusion and large variation in object appearances render these data sets challenging.

*2) Evaluation Metrics:* In MOT challenge benchmark [50], [54], the relationship between ground truth and tracker output is established using bounding boxes. The Intersection over Union (IoU) is used as the similarity criterion, where the threshold is set to 0.5. Then, the tracking performance is measured by MOT accuracy (MOTA, the primary metric), MOT precision (MOTP), the total number of false negatives (FN), the total number of false positives (FP), the total number of identity switches (IDS), the percentage of mostly tracked trajectories (MT), and the percentage of mostly lost trajectories (ML). Specifically, MOTA measures the overall tracking performance of an approach, combined with FN, FP, and IDS. Furthermore, average precision (AP) is adopted to directly measure the prediction accuracy of events.

*3) Subnetwork Implementation:* The proposed model is implemented as follows. We use Resnet-50 as the backbone network and construct a Siamese network to extract the features of detected boxes. For the conv-LSTM, the number of LSTM units is set to eight and the hidden unit size is 1024. Meanwhile, we use a three-layer MLP to model the evolving function whose input is the concatenation of $t - F\_t_{[i]}$ and $h_i^e$.

TABLE I

Ablation Experiments on the Evolving Function, Activation Function, and Training Loss. Models Are Tested on MOT2016 Data Set

| Evolving Function | MOTA↑ | MOTP↑ | MT↑ | ML↓ |
|---|---|---|---|---|
| Parametric Decaying | 42.3 | 77.4 | 18.8% | 40.8% |
| Neural Network | **44.1** | **82.7** | **23.8%** | **36.0%** |

(a)

| Activation $\sigma(\cdot)$ | MOTA↑ | MOTP↑ | MT↑ | ML↓ |
|---|---|---|---|---|
| Sigmoid | 41.9 | 77.5 | 18.6% | 42.2% |
| Biased Relu | 42.4 | 77.2 | 19.0% | 40.0% |
| Elu | 43.1 | 79.5 | 20.0% | 37.7% |
| Softplus | **44.1** | **82.7** | **23.8%** | **36.0%** |

(b)

| Training Loss | MOTA↑ | MOTP↑ | MT↑ | ML↓ |
|---|---|---|---|---|
| Cross Entropy | 40.5 | 77.8 | 16.2% | 48.0% |
| Mean Squared Loss | 42.7 | 77.3 | 19.1% | 40.8% |
| Log-likelihood | **44.1** | **82.7** | **23.8%** | **36.0%** |

(c)

*4) "Bad" Detections Labeling:* As for noisy object detections, we observe every detection box in the MOT data set and label the boxes whose IoU (with any ground truth) are smaller than 0.5 as noisy detections. We then run the baseline algorithm on the data set and label the mismatched detections as confusing object detections. With the above method, we can locate most of the "bad" detections.

*5) Parameter Settings:* In our experiments, we use the public detection results provided by the MOT16 and MOT17 data sets, so as to have a fair comparison with other MOT methods. In the ablation study (Section V-B), we use four training sequences of MOT16 as training data and the other three training sequences for validation. We use the tracklet-based algorithm introduced in Section III-B as our baseline algorithm for MOT. We also choose deep-simple online and realtime tracking SORT [55] and Tracktor [56] as other baselines for comparison purposes. To train the network, the standard Adam optimizer is chosen with a batch size of 32 and an initial learning rate of 0.001. After every 40 000 iterations, the learning rate is further decayed by 10% of the initial learning rate. The training process terminates after 1 50 000 iterations.

### *B. Ablation Study and Discussion*

*1) Comparison on Different Settings:* In this section, we investigate the impact of some practical modifications introduced in Section IV. All results are shown in Table I.

*Evolving Function:* First, we study the evolving function (cf., Section IV-C), and report tracking performances based on different choice of implementation. Results indicate that the neural network model clearly outperforms the parametric decaying function. This verifies our previous hypothesis that the model capacity of the neural network is much larger than the simple parametric evolving function. This helps the model to capture more complex dynamic patterns. Therefore, the evolving function is implemented using the neural network for subsequent studies.

TABLE II

ABLATION EXPERIMENTS ON DIFFERENT COMPONENTS. AP IS ADOPTED TO MEASURE THE PREDICTION ACCURACY

| | Time Independent | Sync RNN | Sync/Async RNN |
|---|---|---|---|
| AP | 26.9% | 27.3% | **35.4%** |
| Speed/FPS | **61** | 43 | 37 |

TABLE III

ABLATION EXPERIMENTS ON DIFFERENT COMPONENTS. ALL MODELS ARE TESTED ON MOT2016 DATA SET

| Components | MOTA↑ | MOTP↑ | MT↑ | ML↓ |
|---|---|---|---|---|
| Baseline | 38.2 | 77.7 | 16.1% | 49.3% |
| + Time Independent | 39.6 | 77.6 | 16.6% | 49.7% |
| + Sync RNN | 41.6 | 77.1 | 19.0% | 39.3% |
| + Sync/Async RNN | **44.1** | **82.7** | **23.8%** | **36.0%** |
| Deep-SORT [55] | 28.4 | **78.8** | 4.8% | 63.1% |
| + Time Independent | 29.1 | 78.5 | 5.8% | 59.4% |
| + Sync RNN | 29.9 | 78.4 | 6.6% | 57.4% |
| + Sync/Async RNN | **30.4** | 78.2 | **7.2%** | **55.1%** |
| Tracktor [56] | 42.4 | 78.2 | 17.3% | 40.4% |
| + Time Independent | 42.9 | 78.0 | 17.7% | 39.9% |
| + Sync RNN | 43.9 | 78.3 | 18.5% | 39.1% |
| + Sync/Async RNN | **44.5** | **78.5** | **19.2%** | **38.6%** |

*Activation Function:* Second, we study the activation function (cf., Section IV-C) and report tracking performances using different $\sigma$ values. As explained earlier, we argue that the intensity function should be non-negative and almost linear when the input is much greater than 0. Thus, the observation that the sigmoid function has the worst result validates our point. Note that the biased Relu function ($\epsilon + \max\{x, 0\}$) satisfies these conditions but performs much worser than softplus function. This is likely because the gradient of the Relu function is strictly eliminated when $x$ is smaller than 0. Thus, the model also trained nonoptimally when the biased Relu is adopted as the activation function.

*Training Loss:* Third, we study the impact of training loss in the optimization of the network and report the results. We observe marginal benefits using log-likelihood function over mean squared error function and cross-entropy loss function. So, we adopt the log-likelihood as our choice of the objective function in all the following studies.

*2) Comparison on Different Components:* We compare the baseline tracking algorithm to the proposed point process enhanced tracking algorithm. The experiments are performed on MOT2016 with different components of our approach. The test AP and speed of point process with different components on MOT16 are shown in Table II, and the MOT metrics are shown in Table III. To outline the superiority and generalization ability of the proposed method, we evaluated the point process model on other tracking methods [55], [56]. Fig. 5 provides visuals of some tracking results by the baseline tracking method and the point process enhanced tracking method. For reference, the speed of our baseline method is 59 FPS on the MOT17 testing data and when it is combined with Sync/Async RNN, the speed is 26 FPS. It is worth noting that the entire procedure is actually a two-step process—the point process method first removes "bad" detections in the sequence, and thereafter the remaining detections are used to construct the tracklets. As such, the point process does not affect the tracking speed. Therefore, we only compare the speed of the point process step using different components.

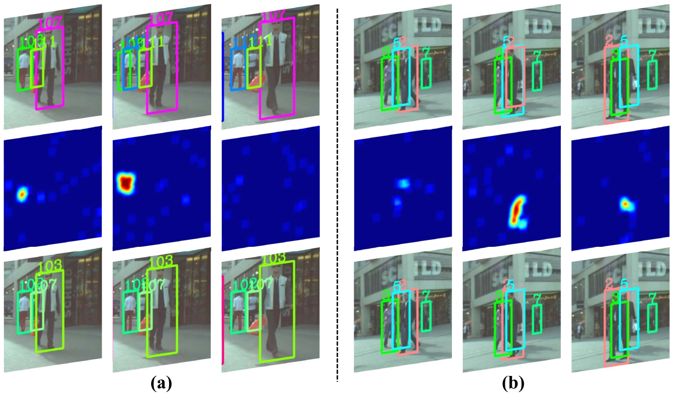


Fig. 5. Two tracking examples. (First row) Results by baseline tracking algorithm. (Second row) Intensity maps generated from the proposed point process model. (Third row) Results by our proposed method. (a) Example of noisy detection. The noisy detection occurs in the second frame of the top row (green box), which causes two different trajectories for one object. (b) Example of confusing detection. The confusing detections occur in the second frame of the top row (blue and red boxes), which causes a mismatch error (identity switch).

In this comparison, the following tracking algorithms are evaluated upon to demonstrate the generalization ability of our approach.

*Baseline:* The tracklet-based algorithm which is introduced in Section III-B. We apply k-means clustering algorithm to generate reliable tracklets followed by softassign [11] algorithm to associate the tracklets into trajectories. For simplicity, we use the terminal detection in each tracklet to calculate tracklet-wise similarity. The baseline algorithm includes a simple threshold-based strategy to filter out noisy detections with low confidences.

*Deep-SORT [55]:* SORT is a pragmatic approach to MOT with a focus on a simple yet effective algorithm. The deep-SORT variant integrates appearance information to improve the performance of SORT. It learns a deep association metric on large-scale person reidentification data set in the offline pretraining stage. Then, during the online application, it establishes measurement-to-track associations using nearest neighbor queries in the visual appearance space.

*Tracktor [56]:* Tracktor is a tracker without any training or optimization on tracking data. It exploits the bounding box regression of an object detector to predict the position of an object in the next frame. It has good extensibility and performs well on three multiobject tracking benchmarks by extending it with a straightforward reidentification and camera motion compensation.

The following components are evaluated to demonstrate the effectiveness of our approach.

*Time Independent:* In this method, we only use the weight-shared CNN to handle the time series input. In other words, at each step, the current frame is input into a fully convolutional neural network (FCN) [69] to generate the intensity map for event prediction. Note that this method has a "memoryless"

TABLE IV
TRACKING PERFORMANCES OF OUR APPROACH AND STATE-OF-THE-ART METHODS ON MOT2016

| Method | MOTA↑ | MOTP↑ | MT↑ | ML↓ | FP↓ | FN↓ | IDS↓ |
|---|---|---|---|---|---|---|---|
| OICF [57] | 43.2 | 74.3 | 11.3% | 48.5% | 6651 | 96515 | 381 |
| CBDA [12] | 43.9 | 74.7 | 10.7% | 44.4% | 6450 | 95175 | 676 |
| QCNN [58] | 44.1 | 76.4 | 14.6% | 44.9% | 6388 | 94775 | 745 |
| STAM [19] | 46.0 | 74.9 | 14.6% | 43.6% | 6895 | 91117 | 473 |
| MDM [59] | 46.3 | 75.7 | 15.5% | 39.7% | 6449 | 90713 | 663 |
| NOMT [35] | 46.4 | 76.6 | 18.3% | 41.4% | 9753 | 87565 | **359** |
| JGD-NL [60] | 47.6 | 78.5 | 17.0% | 40.4% | **5844** | 89093 | 629 |
| TSN-CC [24] | 48.2 | 75.0 | **19.9%** | **38.9%** | 8447 | 85315 | 665 |
| LM-PR [61] | 48.8 | **79.0** | 18.2% | 40.1% | 6654 | 86245 | 481 |
| **Ours** | **50.5** | 74.9 | 19.6% | 39.4% | 5939 | **83694** | 638 |

mechanism, which does not capture any temporal relations within the input data or between event sequences.

*Synchronous RNN:* This method, which is the partial version of our proposed method, takes only the time series as input data in order to reflect the state of the environment where events happen. Correspondingly, the synchronous RNN models the exogenous intensity in our point process.

*Synchronous + Asynchronous RNN:* This is the full version of our proposed method. It includes both synchronous and asynchronous RNN, where the time series and associated event sequence are taken as input data. We adopt a neural network-based time-evolving mechanism to align and merge these two features, for the generation of the intensity maps.

From Tables III, II, and Fig. 5, we make the following observations.

1) The time-independent prediction method outperforms our baseline tracking algorithm. Tracking examples in Fig. 5 shows that directly associating the detected objects or simply using general tracklet association algorithms is ineffective and easily interfered by noisy or confusing detections. Comparatively, by applying the time-independent model to predict and discard the "bad" detections before the association process, we are able to mitigate these noisy and confusing detections to a reasonable measure of success.
2) The synchronous RNN model has significantly better results compared to the time-independent model. The performance gains come from the fact that a learnable model of larger capacity is employed to capture the historical information to cater for complex temporal dynamics. In addition, we observe that both the time-independent model and synchronous RNN model have lower MOTP values than the baseline algorithm; MOTP measures the precision of size and location of bounding boxes in trajectories. This is likely because these two methods mask out the "bad" detections effectively so that they can improve tracking accuracy such as MOTA. This occurs at the expense of losing some "good" detections to misclassification and masking. These errors occur sparsely in time and can be recovered in the tracklet clustering process using linear interpolation (a common scheme also adopted by our baseline tracking algorithm). The only caution is that bounding boxes generated by linear interpolation are less precise than the detector results especially when the objects exhibit large or highly dramatic motions. Nevertheless, we observe that the overlaps between interpolated boxes and ground-truth boxes are still larger than 50%, which may lower the MOTP score but will not affect the MOTA score.
3) The full version of the proposed method (synchronous + asynchronous RNN) performs much better than all other compared methods. This demonstrates the importance of adopting event sequence in addition to time series as inputs. Intuitively, this helps the model to better capture long-term dependencies of events. From another aspect, the asynchronous RNN also reflects the endogenous intensity, as a complementary part to the exogenous intensity already modeled by synchronous RNN. Besides, Table II also demonstrates the effectiveness of the proposed method in predicting the "bad" detection results.
4) Our proposed method consistently improves all the three methods, including the baseline algorithm, the alternative Deep-SORT [55] tracking algorithm, and the Tracktor [56] algorithm in all metrics, which demonstrates its effectiveness and strong generalization ability.

### *C. Comparison With State-of-the-Art Methods*

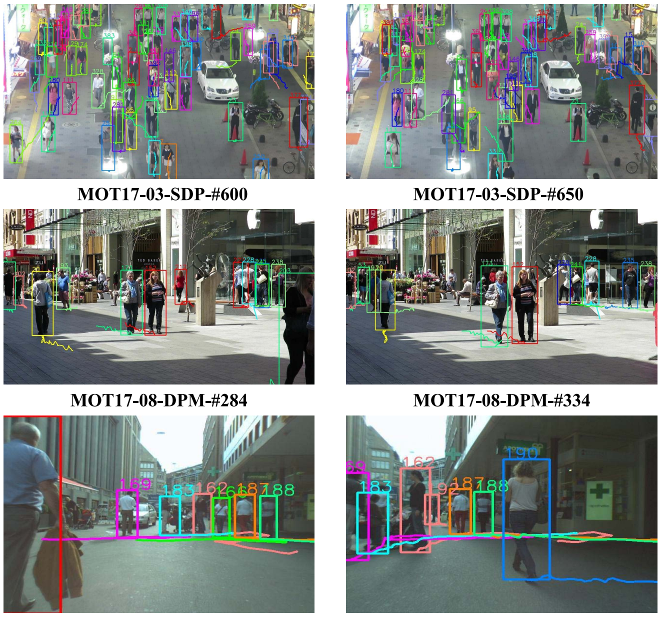
With the best settings of the proposed method affirmed in the ablation study, we conduct further comparisons against state-of-the-art MOT tracker on the widely used benchmark data set: MOT2016 and MOT2017. For fair comparison, all the methods are evaluated and reported based on the same evaluation protocol and metrics. Tables IV and V list the benchmark results for MOT2016 and MOT2017, respectively, comparing the proposed method against recent state-of-the-art MOT trackers such as OICF [57], CBDA [12], QCNN [58], STAM [19], MDM [59], NOMT [35], JGD-NL [60], TSN-CC [24], LM-PR [61], EEBMM [62], NG-bL [63], OGSDL [64], DMAN [65], EDM [21], MHT [66], DLCS [67], CCC [68], and FHFD [17]. Fig. 6 illustrates some tracking results of our method on MOT17 data set.

From the tracking performance comparisons in Tables IV and V, it can be seen that our tracker surpasses all other methods in terms of the primary evaluation metric MOTA. In MOT16, compared to the closest competitor LM-PR [61], our method achieves a 1.7% improvement (50.5%

TABLE V

TRACKING PERFORMANCES OF OUR APPROACH AND STATE-OF-THE-ART METHODS ON MOT2017

| Method | MOTA↑ | MOTP↑ | MT↑ | ML↓ | FP↓ | FN↓ | IDS↓ |
|---|---|---|---|---|---|---|---|
| EEBMM [62] | 44.2 | - | - | - | - | - | **1529** |
| NG-bL [63] | 47.5 | **77.5** | 18.2% | 41.7% | 25981 | 268042 | 2069 |
| OGSDL [64] | 48.0 | 77.2 | 17.1% | 35.6% | 23199 | 265954 | 3998 |
| DMAN [65] | 48.2 | 75.9 | 19.3% | 38.3% | 26218 | 263608 | 2194 |
| EDM [21] | 50.0 | 77.3 | 21.6% | 36.3% | 32279 | 247297 | 2264 |
| MHT [66] | 50.7 | **77.5** | 20.8% | 36.9% | 22875 | 252889 | 2314 |
| DLCS [67] | 50.9 | 76.6 | 17.5% | 35.7% | 24069 | 250768 | 2474 |
| CCC [68] | 51.2 | 75.9 | 20.9% | 37.0% | 25937 | 247822 | 1802 |
| FHFD [17] | 51.3 | 77.0 | 21.4% | **35.2%** | 24101 | 247921 | 2648 |
| **Ours** | **52.4** | 76.6 | **22.4%** | 40.0% | **20176** | **246158** | 2224 |


MOT17-03-SDP-#600

MOT17-03-SDP-#650

MOT17-08-DPM-#284

MOT17-08-DPM-#334

MOT17-06-FRCNN-#454

MOT17-06-FRCNN-#504

Fig. 6. Some tracking results of our method on MOT17 data set.

vs. 48.8%) in MOTA, a 1.4% improvement (19.6% vs. 18.2%) in MT, and a 0.7% improvement (39.4% vs. 40.1%) in ML. In MOT17, compared to the closest competitor FHFD [17], our method achieves a 1.1% improvement (52.4% vs. 51.3%) in MOTA and a 1.0% improvement (22.4% vs. 21.4%) in MT. Note that both LM-PR [61] and FHFD [17] have more sophisticated and computationally heavy tracking pipelines than our baseline algorithm. This implies that we can make significant strides to improve the state-of-the-art by formulating it as a point process method. On the other hand, the MOTP of our approach is slightly lower than some methods because the interpolated detections tend to be less precise when the objects have some large or highly dramatic motions.

Our approach also produces the lowest FN on both MOT16/MOT17 and highest MT in MOT17 (second highest on MOT16), which shows that the proposed method can accurately associate the tracklets and the interpolation process is effective in generating missing detections. Our approach produces the lowest FP on MOT17 (second lowest on MOT16), which demonstrates the strengths of our approach in handling noisy and confusing detections by formulating these "bad" detections using our spatio-temporal point process model.

Although some methods (e.g., CCC [68]) have better performances on the ML and IDS metrics, they obtain this at the expense of sacrificing the performance on other metrics (i.e., MT and FP), which leads to a low overall result in MOTA. Comparatively, our approach can comprehensively balance different aspects in tracking and obtain a better overall result in MOTA. Besides, our approach has relatively high values on some metrics (i.e., IDS) because we use simple features and association schemes to perform tracking. In fact, since our proposed point-process model is generic, we can easily incorporate into our model more sophisticated features and association schemes (e.g., aggregated local flow descriptor in NOMT [35]) to potentially obtain better performances on these metrics.

## VI. CONCLUSION

In this article, we address the issue of misdetections in the MOT task by proposing a novel framework that effectively predicts and masks out noisy and confusing detection results prior to associating objects into trajectories. We formulate the "bad" detection results as a sequence of events, adopting the spatio-temporal point process to model such event sequences. A conv-RNN is introduced to instantiate the point process, where the temporal and spatial evolutions are well captured. Experimental results demonstrate notable performance improvement with the proposed method over state-of-the-art MOT algorithms across different metrics and benchmark data sets.

## REFERENCES

[1] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Subgraph decomposition for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5033–5041.

[2] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6951–6960.

[3] M. Yang, Y. Wu, and Y. Jia, "A hybrid data association framework for robust online multi-object tracking," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5667–5679, Dec. 2017.

[4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[5] S. Gao *et al.*, "Beyond group: Multiple person tracking via minimal topology-energy-variation," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5575–5589, Dec. 2017.

[6] D. Shi, S. Zhang, J. Wang, and Y. Gong, "Detection and association based multi-target tracking in surveillance video," in *Proc. IEEE Int. Conf. Multimedia Big Data*, Apr. 2015, pp. 377–382.

[7] Q. Yu, G. Medioni, and I. Cohen, "Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[8] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. CVPR*, Jun. 2011, pp. 3457–3464.

[9] P. Emami, P. M. Pardalos, L. Elefteriadou, and S. Ranka, "Machine learning methods for solving assignment problems in multi-target tracking," 2018, *arXiv:1802.06897*. [Online]. Available: http://arxiv.org/abs/1802.06897

[10] L. Jin, T. Wu, F. Liu, and G. Zeng, "Hierarchical template matching for robust visual tracking with severe occlusions," *ZTE Commun.*, vol. 10, no. 4, pp. 54–59, 2012.

[11] B. Wang *et al.*, "Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–8.

[12] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[13] J. Xiang, G. Zhang, J. Hou, N. Sang, and R. Huang, "Multiple target tracking by learning feature representation and distance metric jointly," 2018, *arXiv:1802.03252*. [Online]. Available: http://arxiv.org/abs/1802.03252

[14] 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.

[15] X. Zhou, P. Jiang, Z. Wei, H. Dong, and F. Wang, "Online multi-object tracking with structural invariance constraint," in *Proc. BMVC*, 2018, p. 203.

[16] R. Hu, X. Wang, Y. Zheng, and Z. Peng, "Moving target detection and tracking for smartphone automatic focusing," *ZTE Commun.*, vol. 15, no. 1, pp. 55–60, 2017.

[17] 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.

[18] 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.*, 2016, pp. 36–42.

[19] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4836–4845.

[20] Q. He, J. Wu, G. Yu, and C. Zhang, "SOT for MOT," 2017, *arXiv:1712.01059*. [Online]. Available: http://arxiv.org/abs/1712.01059

[21] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 18–27.

[22] A. Taalimi and H. Qi, "Robust multi-object tracking using confident detections and safe tracklets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 1638–1642.

[23] W. Lin *et al.*, "A tube-and-droplet-based approach for representing and analyzing motion trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1489–1503, Aug. 2017.

[24] J. Peng *et al.*, "Tracklet siamese network with constrained clustering for multiple object tracking," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.

[25] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes: General Theory and Structure*, vol. 2. Springer, 2007.

[26] E. Bacry, I. Mastromatteo, and J.-F. Muzy, "Hawkes processes in finance," *Market Microstruct. Liquidity*, vol. 1, no. 1, Jun. 2015, Art. no. 1550005.

[27] J. Yan *et al.*, "Towards effective prioritizing water pipe replacement and rehabilitation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2931–2937.

[28] Ş. Ertekin, C. Rudin, and T. H. McCormick, "Reactive point processes: A new approach to predicting power failures in underground electrical systems," *Ann. Appl. Statist.*, vol. 9, no. 1, pp. 122–144, Mar. 2015.

[29] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song, "Dirichlet-hawkes processes with applications to clustering continuous-time document streams," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2015, pp. 219–228.

[30] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4718–4726.

[31] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Learning time series associated event sequences with recurrent point process networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3124–3136, Oct. 2019.

[32] J. F. C. Kingman, "Poisson processes," in *Encyclopedia of Biostatistics*, vol. 6. Hoboken, NJ, USA: Wiley, 2005.

[33] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[34] V. Isham and M. Westcott, "A self-correcting point process," *Stochastic Processes Appl.*, vol. 8, no. 3, pp. 335–347, May 1979.

[35] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3029–3037.

[36] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Manage. Sci.*, vol. 17, pp. 712–716, Jul. 1971.

[37] X. Gao and T. Jiang, "OSMO: Online specific models for occlusion in multiple object tracking under surveillance scene," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 201–210.

[38] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2874–2883.

[39] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1134–1142.

[40] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, Aug. 2016, pp. 1555–1564.

[41] J. A. González, F. J. Rodríguez-Cortés, O. Cronie, and J. Mateu, "Spatio-temporal point process statistics: A review," *Spatial Statist.*, vol. 18, pp. 505–544, Nov. 2016.

[42] D. Marsan and O. Lengline, "Extending Earthquakes' reach through cascading," *Science*, vol. 319, no. 5866, pp. 1076–1079, Feb. 2008.

[43] E. Choi, N. Du, R. Chen, L. Song, and J. Sun, "Constructing disease network and temporal progression model via context-sensitive Hawkes process," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 721–726.

[44] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Netw.*, vol. 1, no. 4, pp. 339–356, Jan. 1988.

[45] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[47] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[48] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[49] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 876–885.

[50] 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: http://arxiv.org/abs/1603.00831

[51] 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. 2010.

[52] 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.

[53] 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.

[54] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," 2015, *arXiv:1504.01942*. [Online]. Available: http://arxiv.org/abs/1504.01942

[55] 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.

[56] 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.

[57] H. Kieritz, S. Becker, W. Hubner, and M. Arens, “Online multi-person tracking using integral channel features,” in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2016, pp. 122–130.

[58] 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. (CVPR)*, Jul. 2017, pp. 5620–5629.

[59] S. Tang, B. Andres, M. Andriluka, and B. Schiele, “Multi-person tracking by multicut and deep matching,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 100–111.

[60] E. Levinkov *et al.*, “Joint graph decomposition & node labeling: Problem, algorithms, applications,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6012–6020.

[61] 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.

[62] A. Maksai and P. Fua, “Eliminating exposure bias and metric mismatch in multiple object tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4639–4648.

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

[64] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, “Particle PHD filter based multiple human tracking using online group-structured dictionary learning,” *IEEE Access*, vol. 6, pp. 14764–14778, 2018.

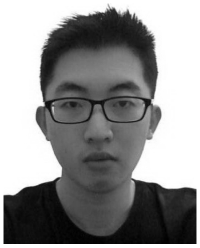
[65] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, “Online multi-object tracking with dual matching attention networks,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 366–382.

[66] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple hypothesis tracking revisited,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4696–4704.

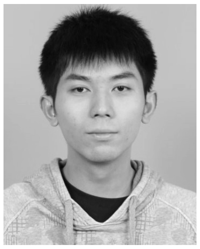
[67] L. Chen, H. Ai, Z. Zhuang, and C. Shang, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, vol. 5, Jul. 2018, pp. 1–6.

[68] 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. 2020.

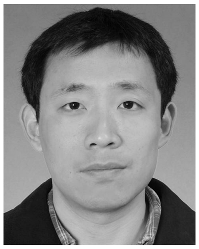
[69] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.



**Tao Wang** received the B.E. degree of electrical engineering and information from Shanghai Jiao Tong University, Shanghai, China, in 2019. He is currently a master student in the Department of Electrical Engineering, Shanghai Jiao Tong University. His research interests include computer vision and machine learning.



**Kean Chen** received the B.E. degree of information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently a master student in the Department of Electronic Engineering, Shanghai Jiao Tong University. His current research interests include computer vision and machine learning.



**Weiyao Lin** (Senior Member, IEEE) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2005, respectively, and the Ph.D. degree from the University of Washington, Seattle, WA, USA, in 2010, all in electrical engineering.

He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. He has authored or coauthored more than 100 technical articles on top journals/conferences including the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, the IEEE Transactions on Image Processing, CVPR, and ICCV. He holds more than 20 patents. His research interests include video/image analysis, computer vision, and video/image processing applications.

Dr. Lin served as an Associate Editor for the IEEE Transactions on Image Processing, the IEEE Transactions on Circuits and Systems for Video Technology, and the IEEE Transactions on Intelligent Transportation Systems. He was an Area Chair of BMVC’2019, ICIP’2019, ACM MM’2018, ICME’2018-2019, and VCIP’2017, and an organizer of more than six workshops in ICME, ICCV, and ECCV. He is a member of the IEEE Multimedia Signal Processing Technical Committee (MMSP TC), a member of the IEEE Visual Signal Processing and Communication Technical Committee (VSPC TC), and a member of the IEEE Multimedia Communications Technical Committee (MMTC). He received the ICME’19 Multimedia Rising Star award.

**John See** (Senior Member, IEEE) received the B.Eng., M.Eng.Sc., and Ph.D. degrees from Multimedia University, Cyberjaya, Malaysia, in 2002, 2005, and 2014, respectively.

He is currently a Senior Lecturer and Chair of the Centre for Visual Computing. He also leads the Visual Processing Lab at the Faculty of Computing and Informatics, Multimedia University. His current research interests cover a diverse range of topics in computer vision and pattern recognition particularly in visual surveillance, image aesthetics, and affective computing. He served as an Associate Editor of EURASIP Journal of Image and Video Processing and a Program Chair of MMSP 2019.

**Zenghui Zhang** (Senior Member, IEEE) received the B.Sc. degree in applied mathematics, the M.Sc. degree in computational mathematics, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2001, 2003, and 2008, respectively.

From 2008 to 2012, he was a Lecturer with the Department of Mathematics and System Science, NUDT. Since 2013, he has been an Associate Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His main research interests include deep learning techniques, target detection and recognition.

**Qian Xu** received the master degree from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2016. She is working as a Computer Vision Researcher at ZTE Corporation, Shenzhen, China. Her research interests are computer vision, deep learning, object detection, and action detection.

**Xia Jia** received the B.Sc. degree from the Taiyuan University of Technology, Taiyuan, China, in 1995, and the M.Sc. degree from the Dalian University of Technology, Dalian, China, in 2001, both in control theory and control engineering. From 2001, she joined ZTE Corporation, Shenzhen, China, and worked in the State Key Laboratory of Mobile Network and Mobile Multimedia Technology. Her main research interests include deep learning techniques, face detection and recognition, Re-ID and activity detection and recognition.