OSMO: Online Specific Models for Occlusion in Multiple Object Tracking under Surveillance Scene

Xu Gao, Tingting Jiang
National Engineering Laboratory for Video Technology, Cooperative Medianet Innovation Center,
School of EECS, Peking University, Beijing 100871, China
{gaoxu1024,ttjiang}@pku.edu.cn

ABSTRACT

With demands of the intelligent monitoring, multiple object tracking (MOT) in surveillance scene has become an essential but challenging task. Occlusion is the primary difficulty in surveillance MOT, which can be categorized into the inter-object occlusion and the obstacle occlusion. Many current studies on general MOT focus on the former occlusion, but few studies have been conducted on the latter one. In fact, there are useful prior knowledge in surveillance videos, because the scene structure is fixed. Hence, we propose two models for dealing with these two kinds of occlusions. The attention-based appearance model is proposed to solve the inter-object occlusion, and the scene structure model is proposed to solve the obstacle occlusion. We also design an obstacle map segmentation method for segmenting obstacles from the surveillance scene. Furthermore, to evaluate our method, we propose four new surveillance datasets that contain videos with obstacles. Experimental results show the effectiveness of our two models.

KEYWORDS

Multiple Object Tracking; Surveillance; Scene Structure Model; Attention-Based Appearance Model; Obstacle Map.

ACM Reference Format:
Xu Gao, Tingting Jiang. 2018. OSMO: Online Specific Models for Occlusion in Multiple Object Tracking under Surveillance Scene. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3240508.3240548

1 INTRODUCTION

Multiple object tracking (MOT) is an important topic in computer vision. There are two kinds of inputs for MOT generally, including the video shot by moving cameras (e.g. ego-motion videos), and the video shot by static cameras (e.g. surveillance videos). However, many recent studies focus on the general MOT that tackle these two types of videos together, and few attention have been paid to surveillance videos specifically. In this paper, we focus on MOT in surveillance videos for two reasons. One is that huge quantities of surveillance videos have been shot everyday. The other is that much prior knowledge can be extracted from the static background in surveillance, which cannot be used in the general MOT.

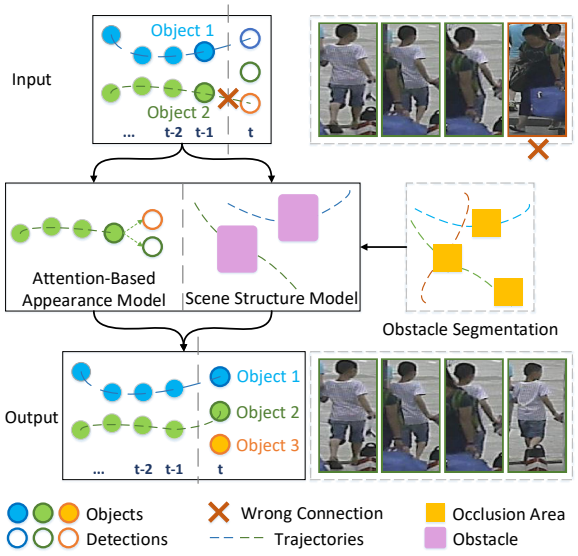

Figure 1: Pipeline of our approach. Input is the tracking result from the baseline MOT method at frame t, and output is the updated tracking result. Those mistracked objects and detections will be matched by optimizing the cost function, which contains our attention-based appearance model and our scene structure model. An obstacle segmentation method is proposed for the scene structure model, which can learn the obstacle map from additional sequences.

Occlusion is the primary difficulty in the surveillance MOT, since it is difficult to track the object when it is occluded. We categorize occlusions in surveillance scene into two types, including the inter-object occlusion and the obstacle occlusion.

The inter-object occlusion is caused by the situation that one object occludes another object, which is also a big challenge in the general MOT. Recent studies have already focused on dealing with this problem. Some approaches try to build robust appearance models to distinguish the two objects before and after the occlusion (e.g. [6, 29, 31]), and [7, 27] also consider the temporal appearance change of these two objects. Besides, some motion models (e.g. [9, 20]) and interaction models (e.g. [27, 35, 41]) are also proposed to deal with the inter-object occlusion.

On the other hand, the obstacle occlusion is caused by the situation that obstacles in the background occlude the object. Since

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240548

recent MOT methods deal with the general MOT, no special care is taken for surveillance videos. As a consequence, these methods could fail at the obstacle occlusion. When the object walks towards the obstacle in the scene and behind the obstacle, these state-of-the-art methods could fail in re-tracking this object after it walks out of the obstacle area. The reason is that general MOT methods do not consider scene structures, thus they cannot distinguish whether its disappearance is caused by an obstacle or by another object. Hence, these scene structures (e.g. obstacle map) are expected to be useful.

Inspired by the above analysis, we proposed an obstacle map segmentation method to describe the scene structure, and two models to deal with occlusions for online multiple object tracking in surveillance. Our obstacle map segmentation method uses additional sequences as input. This is based on the fact that huge quantities of surveillance videos have been shot everyday, and some of them can be used to generate the obstacle map of the scene. For obstacle occlusion, we design a scene structure model with the generated obstacle map as input, which could analyze the relative position between obstacles and missing objects. For inter-object occlusion, based on [27], we design an attention-based appearance model that could measure the appearance similarity between detections and tracked objects by introducing an attention mechanism. The pipeline of our proposed method is illustrated in Fig. 1.

To achieve better performance, we utilize tracking results from the state-of-the-art MOT approaches as inputs frame by frame, and improve these results by optimizing the cost function with our two models. To be specific, two actions are carried out for input trajectories, including "CUT" and "LINK". We "CUT" those trajectories at the current frame if their appearance has a big change, and separate them into the missing object set and candidate detection set. Afterwards, we "LINK" these two sets by optimizing the cost function with our two models. Note that this process is conducted online, and is flexible to refine any state-of-the-art MOT method. Furthermore, to our best knowledge, there exists no benchmark that is dedicated to MOT in surveillance with obstacles. Hence, we build a Surveillance Tracking Benchmark for evaluation, including four brand new and challenging surveillance datasets that contain videos with obstacles. Experimental results on our Surveillance Tracking Benchmark demonstrate the effectiveness of our approach in the surveillance scene compared with state-of-the-art methods.

The main contributions of this paper are as follows:

- An obstacle map segmentation method is designed to segment obstacles from the surveillance scene.
- A scene structure model is proposed to solve the obstacle occlusion in the surveillance scene.
- An attention-based appearance model is proposed to solve the inter-object occlusion.
- Four new challenging surveillance datasets are presented that contain obstacles, and build a new Surveillance Tracking Benchmark for evaluation.

The rest of the paper is organized as follows. In Sec. 2, related works are reviewed. In Sec. 3, 4, 5, we present our obstacle map segmentation method, our scene structure model and our attention-based appearance model respectively. Sec. 6 demonstrates the framework of our MOT method. Details of datasets and experiment results are discussed in Sec. 7. In Sec. 8, we have a concluding remark.

2 RELATED WORKS

Tracking-by-Detection (TBD) methods is popular in recent MOT works [1, 2], which consist of running a detector to generate detections at first and associate those detections afterwards. These methods can be categorized into the offline approach and the online approach. The offline approach generally uses detections from all frames through the entire sequence, followed by a global optimization, such as graph optimization [4, 5, 24, 43] and iterative optimization [32, 33, 38, 42]. Although offline learning approach can achieve a global optimal solution, it cannot satisfy online requirement. The online approach generally uses detections from the current frame as well as previous frames [3, 6, 7, 21, 27, 29, 36, 41]. In this paper, we follow the online MOT strategy.

Appearance Model. Appearance models have been used widely in MOT in order to measure the similarity between the tracked object and the detection, including some simple features such as color histogram [29, 41, 42] and HOG [8]. Recently, deep neural networks have been applied to get appearance models, which use the convolutional neural network (CNN) to extract high-level appearance features [28, 30, 31]. Besides, [27] has taken the temporal information into consideration by using the Long Short-Term Memory network (LSTM), but it has not considered the importance of the object appearance at each different frame.

The appearance model we proposed shares the similar architecture with [27], but differs in two crucial ways: first, we include a new attention sub-network to measure the importance of the object appearance at each different frame. Second, we replace the fully connected layers in [27] with a metric learning network, which refers to the current re-identification work [10].

Obstacle Map Segmentation. The obstacle map is one of the scene structure, which describes the prior knowledge of the scene. Methods to model the scene structure can be generally classified into two categories: scene parsing method and trajectory assistant method. The former one provides a straightforward way to model the scene by recognizing different functional categories, such as trees, buildings and cars [15]. However, the number of functional categories needs to be fixed, which limits the performance. The latter one uses trajectories of objects to generate the scene structure, where the scene structure is more likely to be modeled as an energy map, and areas with high energy are probable to be walkable areas [23, 37, 39, 40]. Nevertheless, these methods can output those targets in the background that can affect the moving pattern of objects, but they cannot output obstacles, which can occlude moving objects.

We follow the trajectory assistant method for generating the obstacle map, and it is utilized to formulate our scene structure model. To our best knowledge, this is the first paper that tackles both MOT and obstacle map segmentation in surveillance.

3 OBSTACLE MAP SEGMENTATION

Inspired by previous trajectory assistant methods, we design a new obstacle map segmentation method. Given the tracking results of additional sequences by the existing MOT method as inputs, we can get an obstacle map for the surveillance scene by analyzing the trend and duration of these trajectories. This is applicable for surveillance application, since massive videos are shot by surveillance cameras. Some of them can be utilized to generate the obstacle map, and

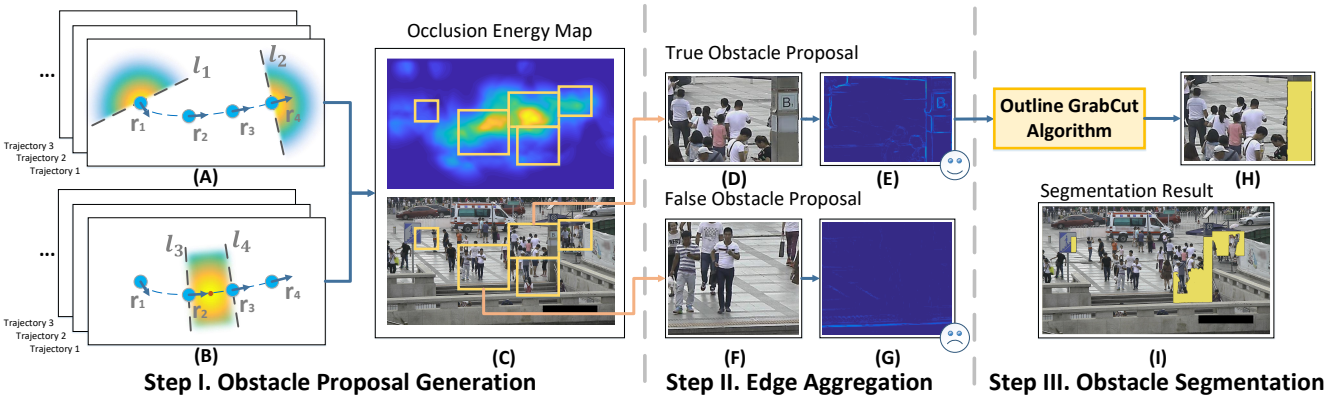

Figure 2: Pipeline of our obstacle map segmentation method from trajectories. Picture (I) shows the final segmentation result of this pipeline.

others for testing MOT performance. Figure 2 demonstrates the pipeline of building the obstacle map for the surveillance scene. Three steps are conducted, including obstacle proposal generation, edge aggregation and obstacle segmentation.

3.1 Obstacle Proposal Generation

Inspired by previous works [23, 37, 39, 40], we first generate an occlusion energy map and then find obstacle proposals from the map, which represent the hypothesis positions of obstacles. Inputs of this step are given trajectories, and outputs are obstacle proposals.

We first build an occlusion energy map from the given trajectories. Suppose that each given trajectory has three states, including *appear*, *disappear* and *reappear*. The *appear* and *disappear* states describe the start and the end of the trajectory, and the *reappear* state describes when the object is lost in one frame and is rediscovered after several frames. In Fig. 2 (A)&(B), $r_1 -> r_2 -> r_3 -> r_4$ is an object trajectory, where r_1 is under the *appear* state, r_3 is under the *reappear* state, and r_4 is under *disappear* state. The object is missing when it is at r_2 position.

We use M_a^i and M_d^i to represent the obstacle energy map of the object o_i under *appear* and *disappear* states. Relying on the fact that objects tend to start from and end at occlusion areas, we use one-sided bivariate Gaussian distributions to model the probability distributions of obstacles around r_1 and r_4, as shown in the Fig. 2 (A). The borderlines l_1 and l_2 are perpendicular to the velocity direction of the object at r_1 and r_4 respectively.

M_r^i is denoted as the obstacle energy map of the object o_i under *reappear* state. Since an occlusion area is expected to exist between r_2 and r_3, we use a restricted Gaussian distribution between r_2 and r_3, which is centered at the middle point of the line segment $\overline{r_2 r_3}$, as shown in the Fig. 2 (B). The restricted borderline l_3 and l_4 are perpendicular to the velocity of the object at r_2 and r_3 respectively.

The occlusion energy map of the object o_i can be derived from the aggregation of M_a^i, M_d^i and M_r^i. Hence, the final occlusion energy map M of the scene can be calculated by

$$M = \sum_{i=1}^{|O_t|} (M_a^i + M_d^i + M_r^i), \tag{1}$$

where $|O_t|$ is the total number of given trajectories.

Once we get the occlusion energy map M, several areas with higher energy can be extracted. We search the occlusion energy map and find bounding boxes that are centered at points with local maximum energy, which are called obstacle proposals. The whole obstacle proposal set is denoted by P. These obstacle proposals are illustrated by the yellow bounding boxes in Fig. 2 (C).

3.2 Edge Aggregation

Once we get obstacle proposals, we use edge aggregation to distinguish whether they truly contain an obstacle or not. Inputs of this step are those obstacle proposals, and outputs are the subset of proposals that are labeled as true obstacle proposal, denoted by P^T. The subset of those proposals that are labeled as false obstacle proposal is denoted by P^F.

The main idea of our edge aggregation method is that textures of obstacles are often static, and textures of non-obstacle areas are unstable, due to the fact that pedestrians or cars would move through this area frequently. Therefore, we calculate the oriented edge map of each obstacle proposal. The aggregation of these edge maps can enhance static edges and cancel those edges with different directions out. As a consequence, those edges that still remain after aggregation are expected to be edges from obstacles, and those edges from non-obstacle areas are expected to be cancelled out.

We use [11] to compute edge maps for the obstacle proposals. The oriented edge map at frame t is denoted as $E(t)$. The aggregated edge map $\mathcal{E}(t_0, t_c)$ between the first frame t_0 to the current frame t_c can be computed by $\mathcal{E}(t_0, t_c) = \sum_{t=t_0}^{t_c} E(t)$.

Ideally, those edges from the aggregated edge map $\mathcal{E}(t_0, t_c)$ are from obstacles after long-term aggregation. If all edges from $\mathcal{E}(t_0, t_c)$ have low intensity, there is probably no obstacle in this proposal. For example, in Step II of Fig. 2, if edges in the aggregated edge map (Fig. 2 (D)&(E)) still remain, this obstacle proposal can be a true obstacle proposal. In comparison, if all values from the aggregated edge map (Fig. 2 (F)&(G)) have low magnitude, this obstacle proposal may be a false obstacle proposal. Those obstacle proposals that survive the edge aggregation will be considered for obstacle segmentation in Step III.

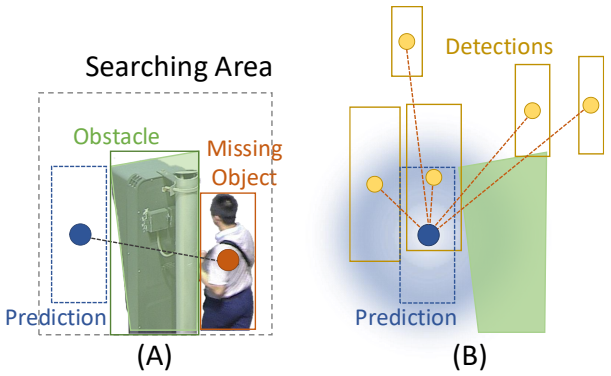

Figure 3: Illustration of our scene structure model: (A) The spatial relation between the obstacle (green quadrangle), the obstacle box (the rectangle that surround the obstacle), the searching area (gray dash rectangle), the missing object (brown rectangle) and the prediction (blue dash rectangle) of the missing object at the frame t_l. (B) The assignments (brown dash line) between the prediction and detections (yellow rectangle) at the frame t_p. A Gaussian distribution centered at the prediction is used to calculate the cost between the missing object and each detection.

3.3 Obstacle Segmentation

After edge aggregation, those obstacle proposals that are labeled as true obstacle proposals P^T are ready to be segmented. Input of this step is P^T, and output is the obstacle segmentation result. The categories of obstacles are uncertain. So we cannot use segmentation methods that are based on known categories. Hence, to take advantage of the aggregated edge map of the obstacle, we choose weakly supervised method with edges as supervision. An outline GrabCut Algorithm [25] is utilized to segment obstacles, where obstacle outline is represented by the aggregated edge map of the obstacle proposal from P^T.

However, there exist objects that seldom move (i.e. a pedestrian is static for a while and then move). These objects may be regarded as obstacles, since edges of static objects are enhanced during the edge map aggregation. Hence, we use an obstacle updating method to eliminate the affect of unstable edges. We denote that the segmentation results of l-th obstacle proposal at frame t is $\Omega_l(t)$, and we merge all $\Omega_l(t)$ into $\Omega(t)$ to get an obstacle map of the whole image. The segmentation result $\Omega(t_0, t_c)$ from the first frame t_0 to frame t_c can be calculated by the product of the segmentation from each frame, which can be formulated by $\Omega(t_0, t_c) = \prod_{t=t_0}^{t_c} \Omega(t)$. The obstacle map segmentation process is continued until $t_c > t_\gamma$ and $|\Omega(t_0, t_c + 1) - \Omega(t_0, t_c)| < \gamma$, where t_γ and γ are hyperparameters. Note that all objects that do not belong to the obstacles are expected to move in $[t_0, t_\gamma]$, thus those objects that seldom move cannot be regarded as obstacles.

As discussed in this section, we take the given trajectories as input, and output the obstacle segmentation map, which is shown in Fig. 2 (I). Afterwards, we use this map as the prior knowledge of our scene structure model.

4 SCENE STRUCTURE MODEL FOR OBSTACLE OCCLUSION

Once we get the obstacle map from Sec. 3, the scene structure model is designed to solve the obstacle occlusion problem. When one object has been occluded by an obstacle, there is high probability that the missing object reappears at the opposite position relative to the obstacle. This is useful prior knowledge for MOT. Hence, our scene structure model takes objects, detections and the obstacle segmentation map as inputs, and output a scene structure cost between each missing object and each detection.

We first give some notions. At the current frame t, **trajectory** denotes a trace formulated by bounding boxes from all the frames between frame 1 and frame $t-1$. **Detection** is a bounding box in frame t, and the detection set is denoted by $\mathcal{D}_t$. **Object** denotes the last bounding box on one trajectory. Objects before frame t comprise the object set for frame t, which is denoted by O_t.

Let us suppose that $|\mathcal{OB}|$ is the number of obstacles, and each obstacle is a polygon. A rectangle is used to exactly surround each obstacle, which is denoted as the obstacle box. We scale the obstacle box into a larger one, as shown in Fig. 3(A). This rectangle is called the *Searching Area* (*SA*). Each obstacle corresponds to one *SA*.

There are two situations for predicting the positions of missing objects. If the object is missing outside all *SA*s of obstacles, it is more likely to be occluded by another object. Hence, we predict the position of the missing object at current frame t as m_i by using a linear motion model assumption. If the object is missing in one *SA*, it is more likely to be occluded by an obstacle. Therefore, we predict the position of the missing object at frame t_p as p_i when the object has walked through the obstacle. To calculate p_i, assume that the i-th missing object o_i is hidden by the obstacle q from frame t_l to frame $t_p - 1$, which means that it will reappear at frame t_p. As shown in Fig. 3 (A), we predict the position of p_i by a linear motion model, which satisfied two conditions. One is that o_i and p_i are on the opposite side relative to the obstacle q. The other is that there is no overlap between q and p_i.

Then, as shown in Fig. 3 (B), a Gaussian distribution is used in the center of the prediction p_i, and the cost between the missing object o_i and the j-th detection d_j at frame t can be formulated by

$$F_s(o_i, d_j) = \begin{cases} \frac{(\widetilde{p}_i - \widetilde{d}_i)^2}{\sigma_1^2} + \frac{(t_p - t)^2}{\sigma_2^2}, & \text{if missing object is in one } SA, \\ \frac{(\widetilde{m}_i - \widetilde{d}_i)^2}{\sigma_3^2}, & \text{if missing object is not in any } SA, \end{cases} \tag{2}$$

where "~" denotes the center of the target. σ_1, σ_2 and σ_3 are hyperparameters, where σ_1 and σ_3 are spatial Gaussian variances, and σ_2 is the temporal Gaussian variance.

5 ATTENTION-BASED APPEARANCE MODEL FOR INTER-OBJECT OCCLUSION

To solve the inter-object occlusion, we propose an attention-based appearance model for measuring the appearance similarity between candidate detections and missing objects. The attention-based appearance model has two modules, including feature extraction and metric learning. Inputs of this model are the objects from previous frames and detections from the current frame, and the output is

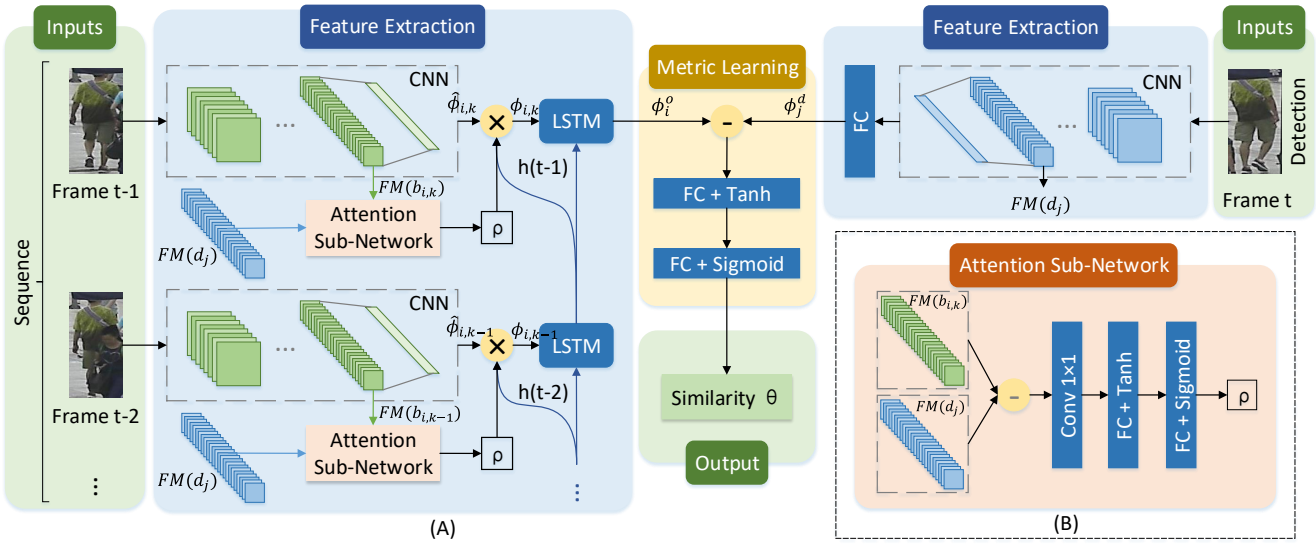

Figure 4: (A) Pipeline of our attention-based appearance model. There are two inputs, including the object trajectory sequence from previous frames, as shown on the left, and the detection at the current frame t, as shown on the top-right. The output is the probability that measures the similarity between each object trajectory and each detection. (B) The detailed structure of the attention sub-network.

the probability that measures the appearance similarity between each object and each detection. The pipeline of the attention-based appearance model is shown in Fig. 4 (A).

5.1 Feature Extraction

Our feature extraction is based on [27], which uses CNNs to extract features of the trajectory at different frames and sends them directly into an LSTM. However, when the candidate detection is partially occluded, the similarity between the candidate detection and the object is likely to decrease, which could lead to a wrong association between the detection and the missing object. Hence, we design an attention sub-network to measure the similarity between the candidate detection and each bounding box on the trajectory T_i. Bounding boxes that are similar to the candidate detection will account for a large proportion of the input of each LSTM cell, which could increase the appearance similarity between the object and the positive occluded detection.

Given a missing object o_i and a candidate detection d_j at frame t, their features can be extracted, where ϕ_i^o and ϕ_j^d denote features of o_i and d_j, and T_i denotes the trajectory of o_i. As shown in Fig. 4, we send the detection d_j into a convolutional neural network (CNN), and the output feature of the CNN will pass through a fully connected (FC) layer to get ϕ_j^d. Meanwhile, the feature of the missing object is extracted as well. We use the same CNN to extract the deep feature of each bounding box on the object trajectory T_i. $\widehat{\phi}_{i,k}$ denotes the output feature of the k-th bounding box $b_{i,k}$ on trajectory T_i at frame t_k, which is derived from the CNN.

Then we use an attention sub-network to measure the similarity between the candidate detection and each bounding box on the trajectory T_i. As shown in Fig. 4 (B), given feature maps of the detection and the missing object as $FM(d_j)$ and $FM(b_{i,k})$ respectively, an element-wise minus is conducted to generate the dissimilarity map, followed by a convolution layer with a kernal size of 1×1 and two FC layers. A sigmoid function is placed at the end to produce the probability ρ. Feature maps of the detection and the missing object are the output of the last convolution layer of the CNN.

Thus, given the similarity probability ρ between the candidate detection d_j and each bounding box on T_i, the feature $\phi_{i,k}$ of the k-th bounding box on T_i can be calculated by a mixture of $\widehat{\phi}_{i,k}$ and h_{t_k-1} by the similarity probability ρ, which is formulated by

$$\phi_{i,k} = \rho \cdot \widehat{\phi}_{i,k} + (1-\rho) \cdot h_{t_k-1}, \tag{3}$$

where h_{t_k-1} is the output of the LSTM cell at frame $t_k - 1$. Features of all bounding boxes from the same trajectory are sent into an LSTM framework. Hence, as shown in Fig. 4 (A), the missing object feature ϕ_i^o is represented by the output of the last LSTM cell.

5.2 Metric Learning

Once we obtain the feature ϕ_i^o of the missing object o_i and the feature ϕ_j^d of the candidate detection d_j, metric learning is utilized to measure the similarity between ϕ_i^o and ϕ_j^d. As shown in Fig. 4 (A), ϕ_i^o and ϕ_j^d are mixed by element-wise minus. The resulting feature can be regarded as the initial dissimilarity feature, followed by two FC layers. A sigmoid function is placed at the end to produce a probability $\theta_{i,j}$ that measures how likely the candidate detection d_j to be similar to the missing object o_i. Hence, the attention-based appearance cost $F_a(o_i, d_j)$ between the object o_i and the detection

d_j can be formulated by

$$F_a(o_i, d_j) = -\log \theta_{i,j}. \tag{4}$$

5.3 Training Strategy

The attention-based appearance model is trained end-to-end. Cross entropy loss is used as the loss function, which can be written as

$$\mathcal{L} = \sum_{i=1}^{|O_t|} \sum_{j=1}^{|\mathcal{D}_t|} -y_{i,j} \log \theta_{i,j} - (1 - y_{i,j}) \log (1 - \theta_{i,j}), \tag{5}$$

where $y_{i,j} = 1$ denotes that o_i and d_j have the same ID, and $y_{i,j} = 0$ denotes the opposite situation. Adam [14] is used to optimize the loss function. Details will be discussed in Sec. 7.2.

6 MULTIPLE OBJECT TRACKING FRAMEWORK

As shown in Fig. 1, we take trajectories that are tracked by the state-of-the-art MOT approaches as input frame by frame, and refine these results by optimizing the cost function with our two models.

Given input trajectories, we operate "WASH" action at first. We find that some state-of-the-art methods would include new bounding boxes that are not detected by the detector, which belong to the negatives. Hence, we "WASH" the input trajectories by removing bounding boxes in the obstacle area that are added by some baseline methods. The reason is that no detection is expected if the object is hidden by an obstacle.

Then a "CUT" action is carried out. For each input trajectory at frame t, we "CUT" them if the appearance similarity between the bounding box at frame t and the trajectory at previous frames is lower than a threshold. The appearance similarity is calculated by our attention-based appearance model. Those "CUT" bounding boxes at frame t comprise the candidate detection set, and those "CUT" objects at frame t comprise the missing object set.

Given the missing object set and the candidate detection set at frame t, the MOT problem turns into an data association problem between two sets, which will be solved by the "LINK" action. As we follow the tracking-by-detection method, the optimal assignment could be solved by minimizing a cost function with two terms, including our appearance term and our scene structure term.

We denote the i-th object at frame t as o_t^i, and the j-th detection at frame t as d_t^j. The state of assignment between object o_t^i and detection d_t^j is denoted as a variable $a_{i,j}$. Here, $a_{i,j} = 1$ describes the situation that detection d_t^j is associated with object o_t^i, and $a_{i,j} = 0$ describes the opposite situation. The assignment set is denoted as $\mathcal{A}_t = \{a_{i,j}\}^{|O_t| \times |\mathcal{D}_t|}$, where $|O_t|$ and $|\mathcal{D}_t|$ are the total number of objects and detections from O_t and $\mathcal{D}_t$ respectively.

Then the optimal assignment set can be formulated by

$$\hat{\mathcal{A}}_t = \text{argmin}_{\mathcal{A}_t} C\Big(O_t, \mathcal{D}_t, \mathcal{A}_t\Big) \tag{6}$$

$$= \text{argmin}_{\mathcal{A}_t} \sum_{i=1}^{|O_t|} \sum_{j=1}^{|\mathcal{D}_t|} a_{i,j} \Big(F_a(o_i, d_j) + F_s(o_i, d_j)\Big) \tag{7}$$

$$s.t. \sum_{i=1}^{|O_t|} a_{i,j} \le 1 \text{ and } \sum_{j=1}^{|\mathcal{D}_t|} a_{i,j} \le 1. \tag{8}$$

Here, $C\Big(O_t, \mathcal{D}_t, \mathcal{A}_t\Big)$ is the cost function, which contains the appearance cost $F_a(o_i, d_j)$ and the scene structure cost $F_s(o_i, d_j)$. The constraints in Eqn. (8) describe that one object should be associated with at most one detection, and one detection should be associated with at most one object. Following the constraints, it is allowed that $\sum_{i=1}^{|O_t|} a_{i,j} = 0$ and $\sum_{j=1}^{|\mathcal{D}_t|} a_{i,j} = 0$, which means that detections cannot be associated with any objects (start a new trajectory), and objects can be missing at current frame. Hungarian algorithm [16] is used to find the optimal assignments.

7 DATASETS AND EXPERIMENTS

7.1 Datasets and Evaluation Metrics

Datasets. Many benchmarks are built for MOT evaluation, such as *2D MOT 2015* [17], *MOT16* [22] and *DukeMTMC* [26]. However, to our best knowledge, there is no benchmark that focuses on the MOT in surveillance scene with obstacles, which is the problem we focus on. Hence, we build a *Surveillance Tracking Benchmark* with obstacles, including 4 challenging surveillance sequences that are first proposed and annotated by ourselves. These new sequences are *NightCrossing*, *MetroOut*, *CrowdedCrossing* and *CampusStone*. Details of our proposed datasets are listed in Table 1.

Table 1: Details of our *Surveillance Tracking Benchmark*.

Dataset	Length	Resolution	Boxes	Obstacle
NightCrossing	1000	1920×1080	27171	Yes
MetroOut	800	1920×1080	25710	Yes
CrowdedCrossing	200	1920×1080	9774	Yes
Campus Stone	1000	1560×1080	1766	Yes

Evaluation Metric. We use metrics proposed in [17] to evaluate the performance in our *Surveillance Tracking Benchmark*. They mainly include MOTA (MOT Accuracy), IDF1 (ID F1 Score), IDP (ID Precision), IDR (ID Recall), MT (Most Tracked), ML (Most Lost), FP (False Positive), FN (False Negative), IDS (ID Switch) and FM (Fragment). MOTA is an essential metric, which combines three sources of error including FP, FN and IDS. IDF1 is another crucial metric, which is the ratio of correctly identified detections over the average number of ground-truth and computed detections. IDP and IDR are fraction of computed detections and ground truth detections that are correctly identified respectively. MT and ML denote the ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% and at most 20% of their respective life span. IDS is the total number of identity switch, and FM is the total number of times a trajectory is fragmented.

7.2 Implementation Details

Details of the attention-based appearance model in Sec. 5 will be discussed. In feature extraction stage, we select ResNet-18 [12] as our CNN model to implement experiments, followed by a 128-dimension FC layer. The hidden layer size of the LSTM is experimentally set as 20, and the maximum input sequence length is set to 20 in consideration of the running efficiency. In the attention sub-network, a 1×1 convolution layer is conducted to the dissimilarity map, followed by two FC layers, and dimensions of these two

Figure 5: Obstacle maps that are generated by our proposed algorithm. Details are discussed in Sec. 7.3.

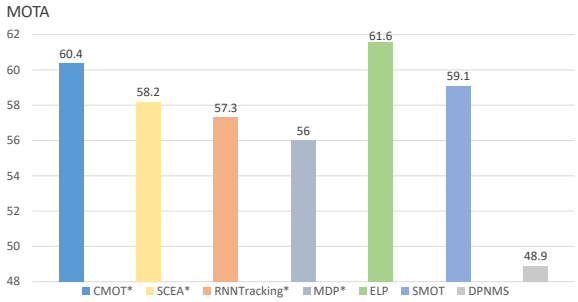

Figure 6: MOTA Comparison of seven state-of-the-art methods in our *Surveillance Tracking Benchmark*. The $*$ denotes that it is an online MOT approach, and without $*$ is an offline MOT approach.

FC layers are set to 128 and 1 respectively. The two CNNs used in feature extraction of both the sequence and the detection share the same parameters. In the metric learning stage, the dimensions of the two FC layers are set to 128 and 1 respectively.

For training strategy, we first choose CUHK03 [18] to pretrain the CNN with the softmax loss. Afterwards, three Re-ID datasets are utilized to pretrain whole network end-to-end, including both the feature extraction stage and the metric learning stage. These three Re-ID datasets are iLIDS-VID [34], PRID2011 [13] and MARS [44], which contain 2658 persons totally, and half of them are used for training and half for testing. For each person, we randomly choose 2 sub-sequences for training, and the sub-sequence length is set to 20. Positives are randomly sampled from the same person, and negatives are selected from another person. Finetuning is conducted after the pretrain process, with three finetuning datasets we built as training sets. The negatives from the finetuning datasets are gathered from the adjacent bounding boxes of the object. These three finetuning datasets are gathered from the same scene as our proposed datasets, which contain 225 persons totally, and all of them are used for training. Note that there is no overlap between datasets for training and datasets for testing.

Besides, in our obstacle map segmentation method, DPNMS [24], which is an offline MOT method, is used as the initialization method to generate input trajectories for obstacle proposal generation. We choose the finetuning datasets we built as training datasets. In the scene structure model, σ_1^2, σ_2^2 and σ_3^2 are set to 3000, 10000, 3000 respectively.

We use detection results from SSD detector [19] as the input of all state-of-the-art methods for evaluating the performance on our *Surveillance Tracking Benchmark*.

7.3 Experiments on Obstacle Map Segmentation Method

We show the results of our obstacle map segmentation method in Fig. 5. Note that some non-obstacle areas have been regarded as obstacles as well. The reason is that some pedestrians stand statically during the entire additional sequence, and their edges will be maintained after the edge aggregation in Sec. 3. This is expected to be solved by shooting additional sequences with a longer period of time. Besides, some obstacles does not been segmented completely, since objects are not dense enough to reveal the obstacle completely. Hence, our obstacle map segmentation method is suitable for the surveillance scene with dense moving objects.

7.4 Experiments on Surveillance Tracking Benchmark

To test the performance of existing methods in our *Surveillance Tracking Benchmark*, we choose seven state-of-the-art approaches that have released open sources, including **SCEA** [41], **CMOT** [3], **MDP** [36], **RNNTracking** [21], **ELP** [20], **SMOT** [9] and **DPNMS** [24]. The former four approaches are online methods, and the latter three approaches are offline methods. The MOTA results of these seven methods are ranked and shown in Fig. 6.

Since our appearance model and scene structure model can be conducted online, we choose the top two online approaches with higher scores in MOTA metric (CMOT and SCEA) as our baselines. We also choose an offline approach that is the best in MOTA metric (ELP) to prove the generalization of our models, which shows our method is useful in both online and offline methods. For each picked baseline method, we conduct two experiments. 'Method + A' does not consider the scene structure model, which means that we assume all missing objects are occluded by other objects. 'Method + A + S' considers both the attention-based appearance model and the scene structure model. The results are shown in Table 2. We can see that in all three baseline methods, using the attention-based appearance model can promote the MOTA score, which illustrates that our appearance model can improve the MOT performance in general. Besides, using the scene structure model can promote the IDF1 score, which infers that our scene structure model can improve the ratio of correctly identified detections, especially under obstacle areas. In addtion, both our two models can decrease ID Switches. Note that the approach which includes both two models achieves the best in MOTA, IDF1, IDP, IDR, MT, ML and IDS with both online and offline MOT approaches as baseline.

To verify the effectiveness of our scene structure model, we demonstrate the efficiency on *CampusStone*, since the major occlusion type in this dataset is the obstacle occlusion. We also manually label the ground truth of the obstacles in *CampusStone* and demonstrate the results on our scene structure model with ground truth

Table 2: Experiments on our *Surveillance Tracking Benchmark*. The 'method + A' shows results that adding attention-based appearance model, and the 'method + A + S' shows results that adding both attention-based appearance model and the scene structure model. The best result of experiments based on each baseline method in each metric is highlight in bold.

Method	MOTA↑	IDF1↑	IDP↑	IDR↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓
CMOT [3]	60.4	50.1	58.7	43.7	32.8%	21.9%	4277	20725	498	821
CMOT + A	**61.0**	50.4	58.9	44.0	**33.6%**	**21.5%**	4197	20458	486	769
CMOT + A + S	**61.0**	**51.0**	**59.6**	**44.6**	**33.6%**	**21.5%**	**4195**	**20456**	**471**	**768**
SCEA [41]	58.2	45.3	58.3	37.0	26.4%	26.0%	**1470**	24983	461	671
SCEA + A	**58.9**	46.3	58.9	38.1	28.3%	**25.7%**	1685	**24415**	369	**601**
SCEA + A + S	**58.9**	**47.4**	**60.4**	**39.0**	**28.7%**	**25.7%**	1676	24444	**332**	603
ELP [20]	61.6	47.2	57.3	40.2	30.9%	**22.6%**	**2413**	21711	639	705
ELP + A	62.8	49.4	58.8	42.6	33.6%	**22.6%**	2837	**20639**	480	611
ELP + A + S	**63.0**	**51.3**	**61.1**	**44.2**	**34.0%**	**22.6%**	2785	20650	**432**	**607**

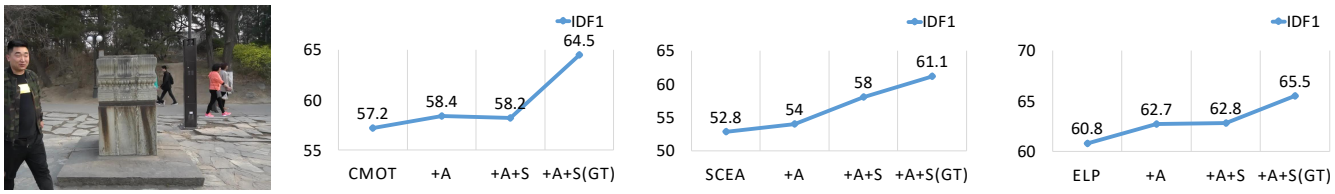

Figure 7: IDF1 metric result in CampusStone Dataset.

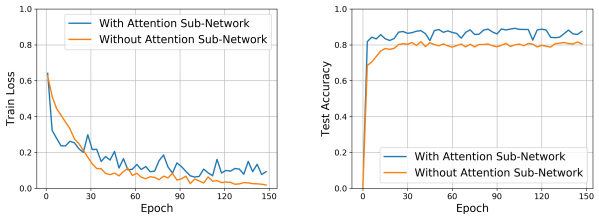

Figure 8: Train loss curve and test accuracy curve on the pre-train process of the attention-based appearance model with and without the attention sub-network.

obstacles. IDF1 curves of all three baseline methods are demonstrated in Fig. 7. For each baseline method, we show four IDF1 results, corresponding to the baseline method, the baseline with our appearance model ('+A'), the baseline with our appearance model and our scene structure model that obstacles, which are generated by either our proposed obstacle map segmentation algorithm ('+A+S') or manually labeled ground truth obstacles ('+A+S(GT)'). From Fig. 7, we can see that our 'method+A+S(GT)' achieves the significant improvement in IDF1 metric, which shows the effect of our scene structure model.

7.5 Experiments of the Sub-Network in Our Attention-Based Appearance Model

To verify the effect of our attention sub-network in the attention-based appearance model, we demonstrate the training loss and the test accuracy of our appearance model in two cases. One is that we use the attention sub-network, as illustrated in Sec. 5. The other one is that we remove the attention sub-network, and send the CNN output of each bounding box from the sequence directly into the LSTM. Figure 8 shows these two comparative models. We can see that the model without the attention sub-network achieves better train loss, but lower test accuracy. On the contrary, the model with the attention sub-network can achieves better test accuracy, which means that the attention sub-network can prevent from overfitting.

8 CONCLUSION

In this paper, we propose two models to solve occlusion for online MOT in surveillance. The attention-based appearance model is proposed to solve the inter-object occlusion, and the scene structure model is proposed to solve the obstacle occlusion. To describe scene structures of the surveillance videos, we propose an obstacle map segmentation method. We also present a new benchmark for MOT in surveillance that contains videos with obstacles, which will improve studies in this area.

ACKNOWLEDGEMENTS

This work was partially supported by National Basic Research Program of China (973 Program) under contract 2015CB351803 and the Natural Science Foundation of China under contracts 61572042, 61390514, 61527804. This work was partially supported by Qualcomm. We also acknowledge the high-performance computing platform of Peking University for providing computational resources.

REFERENCES

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2008. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–8. https://doi.org/10.1109/CVPR.2008.4587583

[2] Anton Andriyenko, Konrad Schindler, and Stefan Roth. 2012. Discrete-continuous optimization for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1926–1933. https://doi.org/10.1109/CVPR.2012.6247893

[3] Seung-Hwan Bae and Kuk-Jin Yoon. 2014. Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1218–1225. https://doi.org/10.1109/CVPR.2014.159

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. 2011. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 9 (2011), 1806–1819. https://doi.org/10.1109/TPAMI.2011.21

[5] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic. 2015. On pairwise costs for network flow multi-object tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5537–5545. https://doi.org/10.1109/CVPR.2015.7299193

[6] Wonggun Choi. 2015. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 3029–3037. https://doi.org/10.1109/ICCV.2015.347

[7] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. 2017. Online Multi-object Tracking Using CNN-Based Single Object Tracker with Spatial-Temporal Attention Mechanism. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 4846–4855. https://doi.org/10.1109/ICCV.2017.518

[8] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 886–893 vol. 1. https://doi.org/10.1109/CVPR.2005.177

[9] Caglayan Dicle, Mario Sznaier, and Octavia Camps. 2013. The Way They Move: Tracking Multiple Targets with Similar Appearance. In *2013 IEEE International Conference on Computer Vision (ICCV)*. 2304–2311. https://doi.org/10.1109/ICCV.2013.286

[10] Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. 2016. Deep Transfer Learning for Person Re-identification. *CoRR* abs/1611.05244 (2016). arXiv:1611.05244 http://arxiv.org/abs/1611.05244

[11] Sam Hallman and Charless C. Fowlkes. 2015. Oriented edge forests for boundary detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1732–1740. https://doi.org/10.1109/CVPR.2015.7298782

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[13] Martin Hirzer, Csaba Beleznai, Peter M. Roth, and Horst Bischof. 2011. Person Re-identification by Descriptive and Discriminative Classification. In *Proceedings of the 17th Scandinavian Conference on Image Analysis (SCIA'11)*. Springer-Verlag, Berlin, Heidelberg, 91–102. http://dl.acm.org/citation.cfm?id=2009594.2009606

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[15] Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. 2012. *Activity Forecasting*. Springer Berlin Heidelberg, Berlin, Heidelberg, 201–214. https://doi.org/10.1007/978-3-642-33765-9_15

[16] Harold W. Kuhn. 2005. Statement for Naval Research Logistics: "The Hungarian method for the assignment problem". *Naval Res. Logist.* 52, 1 (2005), 6–21. https://doi.org/10.1002/nav.20057 Reprinted from Naval Res. Logist. Quart. **2** (1955), 83–97.

[17] Laura Leal-Taixe, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. (2015). arXiv:1504.01942

[18] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 152–159.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016: 14th European Conference on Computer Vision*. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[20] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. 2015. Enhancing Linear Programming with Motion Modeling for Multi-target Tracking. In *2015 IEEE Winter Conference on Applications of Computer Vision*. 71–77. https://doi.org/10.1109/WACV.2015.17

[21] Anton Milan, Seyed Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. 2017. Online Multi-Target Tracking using Recurrent Neural Networks. In *AAAI*.

[22] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A Benchmark for Multi-Object Tracking. *CoRR* abs/1603.00831 (2016). arXiv:1603.00831 http://arxiv.org/abs/1603.00831

[23] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. 2016. Egocentric Future Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4697–4705. https://doi.org/10.1109/CVPR.2016.508

[24] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. 1201–1208. https://doi.org/10.1109/CVPR.2011.5995604

[25] Matthieu Pizenberg, Axel Carlier, Emmanuel Faure, and Vincent Charvillat. 2017. Outlining Objects for Interactive Segmentation on Touch Devices. In *Proceedings of the 2017 ACM on Multimedia Conference (MM '17)*. ACM, New York, NY, USA, 1734–1742. https://doi.org/10.1145/3123266.3123409

[26] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. 2016. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. In *ECCV 2016 Workshop on Benchmarking Multi-Target Tracking*.

[27] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2017. Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 300–311. https://doi.org/10.1109/ICCV.2017.41

[28] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. 2017. Deep Network Flow for Multi-object Tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2730–2739. https://doi.org/10.1109/CVPR.2017.292

[29] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. 2012. Part-based multiple-person tracking with partial occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1815–1821. https://doi.org/10.1109/CVPR.2012.6247879

[30] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. 2017. Multi-object Tracking with Quadruplet Convolutional Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3786–3795. https://doi.org/10.1109/CVPR.2017.403

[31] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. 2017. Multiple People Tracking by Lifted Multicut and Person Re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3701–3710. https://doi.org/10.1109/CVPR.2017.394

[32] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. 2014. Tracklet Association with Online Target-Specific Metric Learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1234–1241. https://doi.org/10.1109/CVPR.2014.161

[33] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. 2017. Tracklet Association by Online Target-Specific Metric Learning and Coherent Dynamics Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 3 (March 2017), 589–602. https://doi.org/10.1109/TPAMI.2016.2551245

[34] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. 2014. Person Re-identification by Video Ranking. In *Computer Vision – ECCV 2014: 13th European Conference on Computer Vision*. Springer International Publishing, Cham, 688–703.

[35] Longyin Wen, Zhen Lei, Siwei Lyu, Stan Z. Li, and Ming-Hsuan Yang. 2016. Exploiting Hierarchical Dense Structures on Hypergraphs for Multi-Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 10 (Oct 2016), 1983–1996. https://doi.org/10.1109/TPAMI.2015.2509979

[36] Yu Xiang, Alexandre Alahi, and Silvio Savarese. 2015. Learning to Track: Online Multi-object Tracking by Decision Making. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 4705–4713. https://doi.org/10.1109/ICCV.2015.534

[37] Dan Xie, Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. 2017. Learning and Inferring "Dark Matter" and Predicting Human Intents and Trajectories in Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, 99 (2017), 1–1. https://doi.org/10.1109/TPAMI.2017.2728788

[38] Bo Yang and Ram Nevatia. 2012. An online learned CRF model for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2034–2041. https://doi.org/10.1109/CVPR.2012.6247907

[39] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2015. Understanding pedestrian behaviors from stationary crowd groups. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3488–3496. https://doi.org/10.1109/CVPR.2015.7298971

[40] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian Behavior Understanding and Prediction with Deep Neural Networks. In *Computer Vision – ECCV 2016: 14th European Conference on Computer Vision*. 263–279. https://doi.org/10.1007/978-3-319-46448-0_16

[41] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. 2016. Online Multi-object Tracking via Structural Constraint Event Aggregation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1392–1400. https://doi.org/10.1109/CVPR.2016.155

[42] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. 2012. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision*. https://doi.org/10.1007/978-3-642-33709-3_25

[43] Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR). 1–8. https://doi.org/10.1109/CVPR.2008.4587584

[44] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. 2016. MARS: A Video Benchmark for Large-Scale Person Re-Identification. In *Computer Vision – ECCV 2016: 14th European Conference on Computer Vision*. Springer International Publishing, Cham, 868–884.