Multiple Object Tracking by Trajectory Map Regression with Temporal Priors Embedding

Xingyu Wan
Xi’an Jiaotong University
Xi’an, Shaanxi, China
wanxingyu@stu.xjtu.edu.cn

Sanping Zhou*
Xi’an Jiaotong University
Xi’an, Shaanxi, China
spzhou@xjtu.edu.cn

Jinjun Wang
Xi’an Jiaotong University
Xi’an, Shaanxi, China
jinjun@mail.xjtu.edu.cn

Rongye Meng
Xi’an Jiaotong University
Xi’an, Shaanxi, China
mengrongye@gmail.com

ABSTRACT

Prevailing Multiple Object Tracking (MOT) works following the Tracking-by-Detection (TBD) paradigm pay most attention to either object detection in a first step or data association in a second step. In this paper, we approach the MOT problem from a different perspective by *directly* obtaining the embedded spatial-temporal information of trajectories from raw video data. For the purpose we propose a joint trajectory locating and attributes encoding framework for real-time, on-line MOT. We firstly introduce a trajectory attribute representation scheme designed for each tracked target (instead of object) where the extracted Trajectory Map (TM) encodes the spatial-temporal attributes of a trajectory across a window of consecutive video frames. Next we present a Temporal Priors Embedding (TPE) methodology to infer these attributes with a logical reasoning strategy based on long-term feature dynamics. The proposed MOT framework projects multiple attributes of tracked targets, e.g., presence, enter/exit, location, scale, motion, etc. into a continuous TM to perform one-shot regression for real-time MOT. Experimental results show that, our proposed video-based method runs at 33 FPS and is more accurate and robust as compared to the detection-based tracking methods and a few other State-of-the-Art (SOTA) approaches on MOT16/17/20 benchmarks.

CCS CONCEPTS

• **Computing methodologies → Tracking**.

KEYWORDS

Multi-Object Tracking; Trajectory Map; Occlusion-aware Radius; Temporal Priors Embedding

ACM Reference Format:
Xingyu Wan, Sanping Zhou, Jinjun Wang, and Rongye Meng. 2021. Multiple Object Tracking by Trajectory Map Regression with Temporal Priors Embedding. In *Proceedings of the 29th ACM Int’l Conference on Multimedia (MM ’21), Oct. 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475304

1 INTRODUCTION

Multiple Object Tracking (MOT) refers to the process to continuously identify, locate and maintain the consistency of multiple targets of interest in consecutive video frames, which has been a fundamental computer vision task for decades. There are three key issues that a MOT framework should handle: 1) Modeling the dynamic motion of multiple targets; 2) Handling the entering/exiting of targets into/from the scene; and 3) Robustness against occlusion and appearance/background variations. Single object tracking tackles 1) and 3) but simply applying multiple single object trackers for MOT [17] usually gives very limited performance due to 2).

Tracking-by-Detection (TBD) method [5, 13] has been a leading paradigm in recent years, whereby the detected bounding boxes of objects in a video sequence are available as prior information, and MOT is then casted as a problem of data association that connects bounding boxes across video frames into trajectories. The tracking performance within the TBD paradigm largely depends on the quality of object detection and data association, both of which get significantly improved by recent Deep Neural Networks (DNN) based approaches [15, 34, 38, 41, 42, 48]. More recently, some researchers [3, 44, 47, 54] attempt to extend the regression-based detector such as Faster R-CNN [36] and Mask-RCNN [15] for one-stage MOT by cascading the bounding boxes regression task to object detection backbones. Since these methods are based on single image object detector, the association process still needs additional clues to operate, and thus the computation is huge, and the performance is limited without end-to-end (i.e., from image-sequence/video to trajectory) capacity.

This has motivated us to investigate if these detection-based methods can be further improved by performing true one-stage MOT that simultaneously locates and tracks targets based on trajectories without any object detection priors. In this paper, we introduce a novel MOT framework that obtains the spatial-temporal

[*] The author is also with the Shunan Academy of Artificial Intelligence, Ningbo, Zhejiang, 315000, China.

MM ’21, October 20–24, 2021, Virtual Event, China.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475304

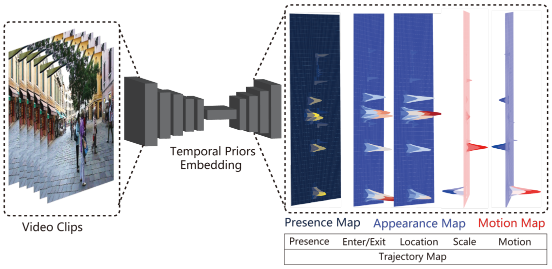

Figure 1: We propose a video-based framework to encode essential trajectory attributes into a multi-channel Trajectory Map (TM) for real-time MOT. The proposed Temporal Priors Embedding (TPE) methodology is designed for simultaneously locating and encoding trajectory attributes of tracked targets through long-term dynamics.

attributes of multiple tracked targets by a single regression framework using a proposed trajectory representation scheme. The challenge is to find suitable representation that is capable of handling both issues 1), 2) and 3) in an online manner, and our idea is to design a multi-task learning framework to jointly locate and encode trajectories by regressing a global Trajectory Map (TM) that describes multiple trajectories, such that for issue 1) and 2), the motion, presence and enter/exit of each target are implicitly represented in the TM, and for issue 3), long-term cues within the window of video clips are utilized by our proposed Temporal Priors Embedding (TPE) process. In this way, our proposed MOT framework can perform one stage visual tracking by directly taking video clips as input and generating multiple trajectory attributes from the obtained TM. Particularly, we propose an Occlusion-aware Radius for calculating the radius of Gaussian kernels when generating ground-truth TM during the proposed TPE process. It is motivated from a fact that too large Gaussian kernels could easily affect other neighboring distributions during occlusions and cause inconsistent distribution problem, while too small ones could result in imbalanced distributions of positive/negative samples during training.

The major contributions of the paper include:

1) A spatial-temporal trajectory-wise representation scheme TM along with an occlusion-aware radius for video-based MOT. Multiple trajectory attributes are encoded into multi-channel TMs including presence/enter/location from presence map, scale from appearance map and motion/exit from motion map, other attributes can be easily extended using this representation;

2) A novel TPE methodology for learning temporal information from video data to maintain positive responses of tracked targets under occlusion, and also to generate positive/negative responses when targets remain/exit the scene;

3) A high-efficient online real-time MOT framework that is capable of one-shot TM regression to jointly locate and encode trajectories without any off-line object detection or re-id feature extraction. It is therefore fast, robust and accurate. Experimental results show that our method runs at 33 FPS with superior accuracy.

2 RELATED WORKS

2.1 Two Stage MOT

This line of works cast the MOT task as separate or cascade processes by associating object detection results into trajectories across video frames. Given a set of detection results as priors, traditional association-based techniques [12, 35] aimed to establish sophisticated models on a frame-by-frame basis. The approaches got improved with better appearance model [21] or more efficient approximation method [14]. Aiming at global optimization with simplified models, the flow network formulations [6, 33, 46] and probabilistic graphical models [1, 30, 50] were considered, along with shortest-path, min-cost algorithms or even graph multi-cut formulations [43]. The major limitation of these methods is the strong prerequisite for high-quality detection results. Trackers adhere to this line of works always need to run detectors first, which result in non-negligible detection times not belong to tracking task itself. Apart from this, to handle imperfect detection, many MOT works in this line operate in off-line, back-and-forward fashion [29, 33, 41, 42] to handle ambiguity, and are therefore computational huge with limited accuracy.

2.2 One Stage MOT

Owing to the rapid progresses of object detection techniques, visual trackers can be further integrated with object detectors to perform one-stage MOT in seek of time and computational efficiency.

Detector-extended Tracking. MOT works derived from image-based object detectors seem to be prevailing recently, because the rapid progresses on detection are rather easy to extend to one-stage multi-object trackers. To give some examples, Tracktor [3] proposed to convert Faster R-CNN [36] to tracktor by cascading bounding boxes regression to region proposals. CenterTrack [54] extended CenterNet [55] to conduct object detection and adjacent frame association in one network. These one-stage methods provided a new inspiration for MOT that, visual tracking could be accomplished within modified detection networks. However, image-based object detector is not optimal to target locating problem in MOT task as the tracked targets can be invisible due to occlusions or variations. One effective scheme is to jointly conduct detection and feature embedding in a unified network by sharing low-level features. Track-RCNN [44] extended Mask-RCNN [15] by adding person re-identification feature to MOT, and JDE [47] adopted YOLOv3 [34] with an extra embedding task. The tracking performance of these methods were still limited, because object detection aims to retrieve positive signals based on spatial information, while embeddings in MOT should engage temporal cues for analyzing long-term variations. The inconsistency between these two tasks makes it hard for an integrated tracker to handle complicated situations especially like long-term occlusions.

Video-based Tracking. To overcome the inconsistency between the spatial and temporal signals, some works propose to conduct Single Object Tracking (SOT) in a video-based way. [20] introduced a spatial-temporal tubelet to increase object detection robustness by incorporating temporal consistency. [10] proposed to explore the local correlation of Region of Interest (ROI)s between adjacent frames with a Siamese Region Proposal Network. These works focus on establishing correlation between spatially detected objects

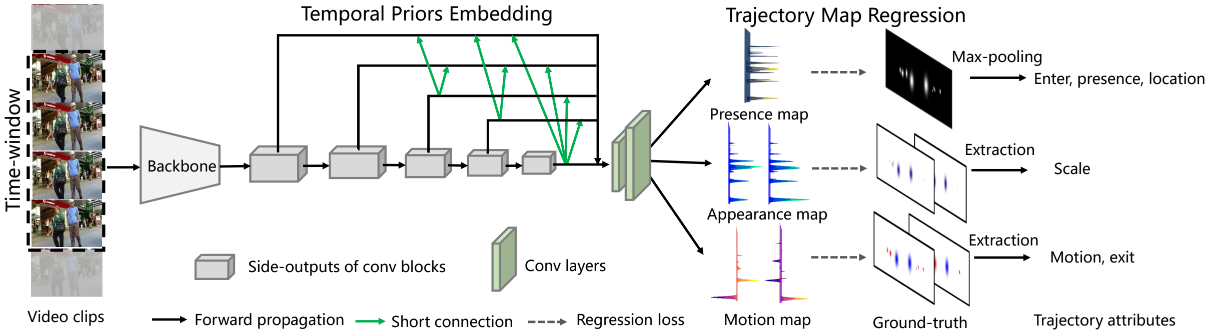

Figure 2: One-stage MOT framework by TM regression. We adopt short connections [18] between each conv block, and decode the feature maps to a multi-channel TM in three branches for multi-task learning. The total network takes a time window of video clips as inputs, and outputs the regressed multi-channel TM. At online tracking, positive responses are retrieved from the presence map by max-pooling, while other trajectory attributes are extracted from the corresponding TM channels, and the final trajectories are generated by concatenating the attributes at each timestamp.

to maintain robust motion prediction for individual target, but still remain challenging when there are multiple targets that exhibit strong dynamics simultaneously.

Recently, a few works attempt to conduct MOT directly from video. CenterTrack [54] proposed to track from adjacent frames with an extra tracking-conditioned heatmap. Since it only focused on improving the consistency of spatial detections, the association performance in [54] is quite limited in dense crowded scenes. TubeTK [31] proposed to adopt bounding tubes to predict motion tendencies in a 3D Convolutional Neural Networks (CNN). It replaced the bounding box with multi-frame bounding tubes and effectively reduced the inconsistent spatial detections during short-term occlusions, but at the cost of larger processing expense and lower time-efficient with the introduced 3D CNN.

Inspired by these, we propose a more general video-based framework to learn spatial-temporal information for jointly locating and encoding trajectories. It not only benefits for video-based detection but also makes the association process extremely easy and robust. Moreover, due to our proposed TPE module, our video-based tracking framework can runs faster than real-time, which is even faster than typical image-based object detector.

Note that our proposed framework is significantly different from detection-based methods like CenterTrack [54] and others [3, 44, 47] in three ways: 1) Our proposed feature scheme TM is designed for the MOT task by encoding multiple trajectory attributes, while CenterTrack [54] is based on extending object detection to describing motion. Each signal on TM represents for an activated trajectory instead of discrete object. Based on this, novel state attributes such as *presence*/*enter*/*exit* and others in the future can be easily extended in our framework and learned end-to-end; 2) Our framework handles key tracking problems such as trajectory states estimation in a video-based way, instead of discrete estimation and detection steps; 3) We evolved a Gaussian like expression of objects from detection to a better version for tracking task by introducing occlusion-aware radius, which is more robust against occlusion than a typical, fixed Gaussian kernel width. This plays an important role for our video-based tracker to achieve better tracking performance and faster running speed than image-based ones at the same time.

Table 1: Encoding trajectory attributes into a multi-channel Trajectory Map.

Trajectory Map	Channel	Encoded Attributes
Presence	1	*enter*, *presence*, *location*
Appearance	2	*scale*
Motion	2	*motion*, *exit*

3 THE PROPOSED ALGORITHM

To accomplish MOT in one-stage forward propagation, our focus is to treat the tracking process by a regression framework, so that multiple continuous trajectories can be obtained on-line without explicit detection, or re-id appearance feature representation. This section depicts the two major components in our MOT approach, i.e., TM as an effective representation scheme, and TPE as an effective training process. The overall structure of our proposed model is shown in Fig. 2, and the following subsections elaborate the system.

3.1 Multi-channel Trajectory Map

A representation scheme is required to encode trajectories so that a DNN based regressor can be trained. To encode sophisticated feature dynamics of interesting targets in tracking scenarios, we define the TM to be a multi-channel feature map for representing different attributes of trajectory. As listed in Table 1, a complete TM consists of three portions: the "presence map", the "appearance map", and the "motion map", to encode six kinds of trajectory attributes, including *presence*, *enter*, *location*, *scale*, *motion* and *exit*. Each attribute is encoded into one or several channels based on the dimension of the stated attribute.

To elaborate, we represent the attribute *presence* in one channel on the "presence map" portion of TM. For each target to be tracked, we represent its presence as a Gaussian-like distribution in $[0, 1]$ with target-wise ellipse shape. The positive value from this presence map means target being tracked “is” activated which is essential for identifying “remain/enter" of target during tracking. Similarly, the attribute $location = \{x, y\}$ representing the center locations of activated targets is naturally encoded in presence map by positive response distributions centering at x, y for use of target locating. The attributes $scale = \{w, h\}$ and $motion = \{dx, dy\}$ are also represented in the corresponding map channel, namely the “appearance map” and “motion map”. Also, attribute *exit* is encoded by the motion map. Given ground-truths tracking data, for trajectories that end at certain time step and never reappear in subsequent video frames, we take them as actually leaving the scene, i.e., “exit”. We set $motion = \{dx, dy\}$ at next time step to $(-x_{end}, -y_{end})$ for these targets, where x_{end} and y_{end} are the last center locations. In this way, if target predicted locations approximate to 0, we can identify them as “exit”. Generally, our TM is a five-channel feature map salient on tracked targets where each attribute of a trajectory is represented as target-wise distributions of response values on the corresponding channels.

3.2 Size-derived Target Distribution

Modified from the standard derivation from CornerNet [24], we employ a size-derived distribution for identifying tracked targets as small radius of positive responses defined by an ellipse Gaussian kernel. Given a bounding box scale $s_k = (w_k, h_k)$ for each target k, we first express size-adaptive kernel radius r_k from a quadratic parabolic equation:

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

Here coefficients $\gamma_1, \gamma_2, \gamma_3$ are all related to s_k for ensuring a minimum overlap $\alpha = 0.7$ with GT. Followed by this, the radius r_k and sigma σ_k of each Gaussian-like distribution are derived as following:

$$r_k = |\frac{\gamma_2 + \sqrt{\gamma_2^2 - 4\gamma_1\gamma_3}}{2\gamma_1}|, \qquad \sigma_k = \frac{r_k}{3} \,. \tag{2}$$

Using derivations above we can obtain size-derived radius r_{kx} and r_{ky} by adopting w_k and h_k both in width and height, and the corresponding σ_{kx}, σ_{ky}. Then we represent an ellipse Gaussian-like distribution for each pixel (i_k, j_k) of target k within a defined ellipse range as following:

$$\phi_k(i_k, j_k) = exp(-\frac{(x_k - i_k)^2}{2\sigma_{kx}^2} - \frac{(y_k - j_k)^2}{2\sigma_{ky}^2}) \,, \tag{3}$$

where $i_k \in [-r_{kx}, r_{kx}]$ and $j_k \in [-r_{ky}, r_{ky}]$. As illustrated in Fig. 3, this size-derived distribution for each activated target is centered at (x_k, y_k), the distribution area is restricted by radius (r_{kx}, r_{ky}), and the dispersion of value is controlled by $(\sigma_{kx}, \sigma_{ky})$.

3.3 Occlusion-aware Radius

Simply adopting the above derivations for each tracked target may lead to inconsistent distributions in value domain when encountering severe occlusions. As shown in Fig. 3, when two centroids of distributions are too close, one positive response could be filtered out after local Non-Maximum Suppression (NMS) and resulting in a False Negativ, or incorrect attribute value could be extracted for one target after TM regression. Existing one-stage methods[26, 47] proposed to add extra embedding heads for a further estimation to help reduce False Negatives and ID Switches. These embeddings were still learned from spatial information which were not sufficient enough to discriminate neighbouring tracklets, not to mention the additional processing expenses introduced to the tracking system. Here we introduce an occlusion-aware radius to ensure the consistency of multiple independent distributions on global TMs over long-term occlusions.

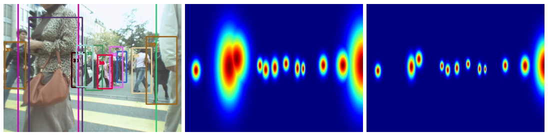

Figure 3: Exemplary of size-derived Gaussian-like distribution with occlusion-aware radius. From left to right: images annotated with gt bboxes, a single channel TM without/adopting occlusion-aware radius.

For a trajectory $X_k^{t-m\sim t}$ with age m at timestamp t, we use the maximum Intersection Over Union (IOU) $IOU_k^t = \max_N IOU^t(k, N)$ with rest N targets to estimate the occlusion status for target k at timestamp t. If $IOU_k^t > 1 - \alpha$, we take target k *is* under occlusion at t, here we adopt the same α as section 3.2 for consistency. Assume target k being under occlusion from $t - l1$ to $t + l2$, we first use $IOU_k^o = max(IOU_k^t|_{t-l1}^{t+l2})$ to identify the most occluded status o over occlusion period $l1 + l2$, and obtain the reduced radius $r_k^o = r_k^{t-l1-1} \times (1 - IOU_k^o)$. Next at each timestamp t during occlusion period $l1 + l2$, the occlusion-aware radius $\tilde{r}_k^t$ is defined as following:

$$\tilde{r}_k^t = r_k^o + a_t \,, \tag{4}$$

$$a_t = r_k^o \times b_t, b_t = 1 - \frac{IOU_k^t}{IOU_k^o} \,. \tag{5}$$

Here a_t is a temporal smoothing item and b_t is a penalty factor for occlusion. Additionally, for targets whose centers still fell into other Gaussian shapes after adopting occlusion-aware radius, we identify them as “too close” when their center point distances are lower than ϵ. We pull them away from each other for $\frac{\epsilon}{2} + 1$ against their moving directions to ensure the completeness of each distribution. Here parameter ϵ is set to 5 from training data in our experiments.

3.4 Temporal Priors Embedding

Targets can be invisible at certain frames due to occlusion or exit, where invisible status of a tracked target does not mean this target actually leaves the scene. Locating and tracking targets based on single frame may not be accurate for the situation, and these are some of the typical cases where temporal information should be used to smooth the extraction of certain trajectory attributes. Hence we propose the TPE process to consider long-term temporal dynamics of tracked targets and obtain video-based spatial-temporal trajectory features, thus to generate more accurate TM.

At each timestamp, the trajectory attributes of tracked targets encoded on the multi-channel TM are learned from a window of history states. Fig. 4 demonstrates the TPE process for generating

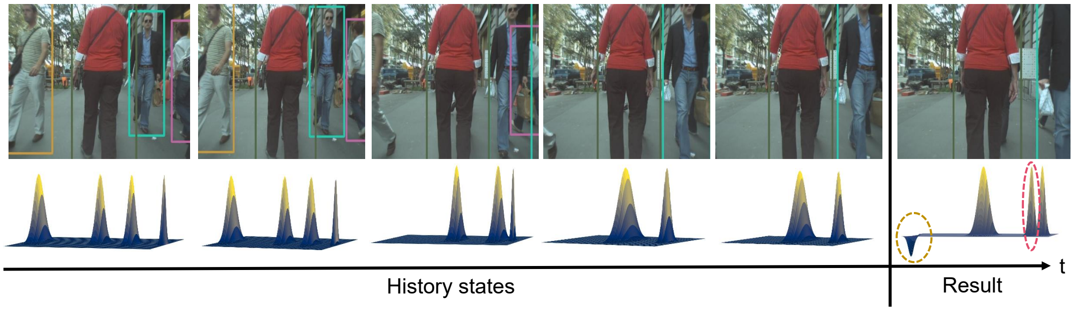

Figure 4: Exemplary of TPE process, best viewed in color. The first row illustrates image sequences annotated by ground-truths, the second row visualizes the corresponding single channel TMs at each time stamp. We use TPE process to maintain positive response for target being occluded (annotated by pink), and give negative response for target actually leaves the scene (annotated by yellow).

an ideal TM, where the utilized prior refers to the historical positive/negative states of any target. For a trajectory $X_k^{t-m\sim t}$ lasting for m frames at time step t, we estimate the current activated status and its corresponding attributes values by conducting a logical reasoning upon a time window of history states $H_k^n = \{h_k^l, l = t-n, t-n+1, t-n+2, ...t-1\}$ with a window length n. The specific logical reasoning method is described as:

$$z_k^t = \begin{cases} s_{pos}, & \text{if } z_k^{t-1} = s_{pos} \text{ or } p_k^t >= \beta, \\ s_{neg}, & \text{otherwise}, \end{cases} \tag{6}$$

$$p_k^t = \frac{\sum_{l=t-n}^{l=t-1} h_k^l}{n}. \tag{7}$$

Here z_k^t denotes the trajectory attributes for target k at time step t. If target was positive at last timestamp, we still maintain a positive response s_{pos} at t. Otherwise we use p_k^t from Eq. (7) to estimate the proportion of positive states within the time window. If the history states h_k^l of target k remain positive during most of the past time, i.e., $p_k^t >= \beta$, we take this target as activated on TM. Conversely, if a target remains negative in history within time window, i.e., $p_k^t < \beta$, we consider it not activated currently and give negative responses for this target on TM. The status of activated or not corresponds to the positive and negative responses $\{s_{pos}, s_{neg}\}$ on all channels of TM, and the actual values of positive-negative responses are different for each channel. Values of $\{s_{pos}, s_{neg}\}$ for presence map are $\{1, 0\}$, for appearance map are $\{w/h, 1\}$, for motion map are $\{dx/dy, 0\}$ (if $enter/presence$) or $\{-x_{end}/-y_{end}\}$ (if $exit$). We use the proposed TPE to 1) generate training labels at each time step, and 2) learn long-term spatial-temporal embeddings for a time-window of video clips as shown in Fig. 2.

3.5 Multi-task Learning Network

As shown in Fig. 2, our TM regression network is an encoder-decoder network adopting short connections proposed in [18] between feature maps of convolutional blocks. The feature encoding branch can be simply embedded with several well-known backbone networks, such as VGGNet [40], ResNet [16], and ResNext [49]. For VGGNet [40], we adopt short connections between five stages of conv blocks to fuse multi-scale feature maps for combining both low-level and high-level information to learn a more inherent representation. The regression branch is a multi-channel decoder for TM generation. Each channel of TM for regressing a particular attribute is optimized by using different loss functions. In general, our MOT network takes a time window of video clips as inputs, and directly outputs the multi-channel TM. The loss functions for optimizing the proposed multi-task regression problem are defined as following:

$$\mathcal{L}_{Presence}(Y_P, \hat{Y}_P) = \frac{1}{\sum_{i=1}^m y_P^i} \sum_{i=1}^m BCE(y_P^i, \hat{y}_P^i), \tag{8}$$

$$\mathcal{L}_{Appearance}(Y_A, \hat{Y}_A) = \frac{1}{\sum_{i=1}^m y_P^i} \sum_{i=1}^m L1(y_A^i, \hat{y}_A^i), \tag{9}$$

$$\mathcal{L}_{Motion}(Y_M, \hat{Y}_M) = \frac{1}{\sum_{i=1}^m y_P^i} \sum_{i=1}^m L1(y_M^i, \hat{y}_M^i). \tag{10}$$

Here $\mathcal{L}_{Presence}$ is the mean element-wise cross entropy loss for regressing presence map, Y_P and $\hat{Y}_P$ indicate the output presence map and ground-truth, while y_P and $\hat{y}_P$ are the corresponding pixel values from Y_P and $\hat{Y}_P$, and m is the total number of positive responses from ground-truth. Similarly, $\mathcal{L}_{Appearance}$ and $\mathcal{L}_{Motion}$ indicate the element-mean L1 loss functions for regressing appearance map and motion map respectively. Our total loss is a weighted fusion of each regression branch.

3.6 Trajectory Generation

As listed in Algorithm 1, the multi-channel TM is firstly output after one-forward network propagation at each timestamp t, then we generate the tracking results by simply extracting attributes from output TM and linking into trajectories using a greedy matching. Specifically, we retrieve attributes *presence* and *location* from the positive distributions on presence map Y_P^t after NMS. Attributes *scale* and *motion* are directly extracted from the corresponding appearance map Y_A^t and motion map Y_M^t using the retrieved *location*. Positive response with attribute *presence* will be identified as *enter* and initialized as a new track in two cases: 1) at the initial frame of testing video; 2) does not match with any existed trajectories at current frame.

```
Input: Tracking results $T_N^{t-1} = \{location, scale, age\}$ and
        network outputs $\{Y_P^t, Y_A^t, Y_M^t\}$.
Output: Tracking results $T_H^t$ at $t$.
Attributes Extraction:
$L_D^t = \{(x_{1\sim D}^t, y_{1\sim D}^t)\} \leftarrow NMS(Y_P^t)$ // extract location;
$S_D^t = \{(w_{1\sim D}^t, h_{1\sim D}^t)\} \leftarrow Y_A^t | L_D^t$ // extract scale;
$M_N^t = \{(dx_{1\sim N}^t, dy_{1\sim N}^t)\} \leftarrow Y_M^t | L_N^{t-1}$ // extract motion;
$O_D^t \leftarrow L_D^t \cup S_D^t$ // group D observations.
for $k = 1; k \leq N; k++$ do
    $\tilde{l}_k^t = (\tilde{x}_k^t, \tilde{y}_k^t) \leftarrow l_k^{t-1} + M_k^t$ // predicted location;
    if $\tilde{l}_k^t <= 1$ or $age_k > L$ then
        Terminate track $T_k^t$ // exit or out of date.
    end
    else if $o_g^t \leftarrow argmin(Dis_G(O_D^t \cap sr_k^t, \tilde{l}_k^t))$ then
        Update $T_k^t \leftarrow o_g^t$; remove $o_g^t$ from $O_D^t$ // match.
    end
    else
        $age_k + 1$ // not match.
    end
end
Initialize new tracks for $o^t \in O_D^t$ // enter.
```

Algorithm 1: Online tracking algorithm

For each target k at timestamp t, we use the extracted attribute *motion* to generate a search region $sr_k^t = \{\tilde{x}_k^t \pm w_k^{t-1}, \tilde{y}_k^t \pm h_k^{t-1}\}$ for matching with observations. Here $(\tilde{x}_k^t, \tilde{y}_k^t)$ are the predicted center locations $\tilde{l}_k^t$ using attributes *location* and *motion*. We adopt a gated distance Dis_G composed of IOU and center point distances to measure the affinity between target k and D observations within the search region. Matched positive response is linked to target for generating the trajectory X_k^t. Attribute *exit* is identified for a target if the predicted location $\tilde{l}_k^t$ is approximate to zero. Targets being identified as *exit* will be terminated at once, while targets not match with any observation will remain in L frames.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

We evaluate our method on MOT16/17/20 Benchmarks [7, 28]. MOT16/17 contain the same video sequences including 7 for training and 7 for testing, but the annotated tracks are different. MOT20 contains 4 videos for training and 4 for testing under crowded scenes with high densities. All these benchmarks provide public detection results from DPM [11], Faster-RCNN [36] and SDP [51] for evaluations under public protocol as well as private detections. Our backbone is first initialized on ImageNet [8] and pre-trained using CrowdHuman [39], then we fine-tune our total network on MOT [7, 28] training sets and use test sets for evaluation. We conduct evaluation following the standard CLEAR MOT metrics [4] along with the IDF1 score [37] as well as other measurements, including Multiple Object Tracking Accuracy (MOTA), ID F1 Score (IDF1), Mostly Tracked Targets (MT), Mostly Lost Targets (ML), False Positives (FP), False Negatives (FN), Identity Switches (ID Sw.), and Fragments (Frag). Main evaluation metrics are composed of MOTA which reflects the coverage of ground-truth trajectories, and IDF1 score which quantifies the robustness of maintaining identities.

Table 2: Ablation studies on MOT17 training set. The best results are highlighted in bold. ↑/↓ indicates higher/lower score denotes better performance.

Method	MOTA↑	IDF1 ↑	FP ↓	FN ↓	ID Sw. ↓
w.o. TPE	62.3	55.7	2,921	2,0564	1,422
w.o. occ	61.3	61.9	2,989	2,3284	1,221
w.o. *motion*	66.1	54.8	2,212	1,4305	1,704
w.o. *exit*	67.0	64.1	2,112	1,4251	931
Ours	**68.3**	**67.6**	**1,910**	**1,4061**	**779**

4.2 Implementation Details

We implemented our MOT framework in Python3.6 using Pytorch 1.7 with 4 NVIDIA RTX 2080S GPUs. The kernel size of local NMS and the confidence score for decoding positive responses at tracking phase (Sec. 3.6) were set to 11 × 11 and 0.3. The threshold value of IOU within the defined gated distance and the length L for remaining unmatched trajectories (Sec. 3.6) were set to 0.3 and 10 respectively. Each image frame within a time window input was resized to 480 × 480 for width and height. The total training epochs were 500 and the learning rate was initialized to $1e-3$ and divided by 10 every $\frac{1}{3}$ epochs. We used Adam [22] for optimizer and the batch size is set to 8. More details of hyper-parameters settings and data augmentations can be found in supplementary material.

4.3 Ablation Study

The ablation studies were conducted on MOT17 training set without any fine-tuning. To start with, Table 2 shows a quantitative result on each major component of our tracking framework.

Temporal Priors Embedding. Without using TPE, our network takes one single frame as input, and the attributes are only inferred spatially without any history prior, i.e., the tracking process is therefore reduced to linking spatially detected object locations. As seen in the first row of Table 2, the clear performance gap from our complete setting shows that, the TPE process is a key component which enables our framework to directly learn to track targets instead of discrete objects.

Occlusion-aware Radius. We introduce occlusion-aware radius (short as "occ") to ensure the consistency of multiple independent distributions on global TMs for our video-based framework. As we show, simply adopting TPE to video-based framework without occ, the tracking performance is even worse than image-based one. This is because occlusions are very common in tracking scenes where targets distributions are easily overlapped with each other, which makes the association model hard to distinguish them. By introducing occlusion-aware radius to our feature representation, the derived Gaussian radius is estimated upon temporal priors and reduced to a reasonable value during occlusions for maintaining target distribution independence.

Motion Map Module The motion map module, as part of TM, is trained as a branch of a multi-task learning network to establish

Table 3: Quantitative comparisons with other alternative components on MOT17 training set.

Method	MOTA↑	IDF1 ↑	FP ↓	FN ↓	ID Sw. ↓
FlowNet2 [19]	67.9	67.4	**1,903**	**1,4061**	935
Hungarian [23]	68.1	**67.7**	1,912	1,4251	**769**
Ours	**68.3**	67.6	1,910	**1,4061**	779

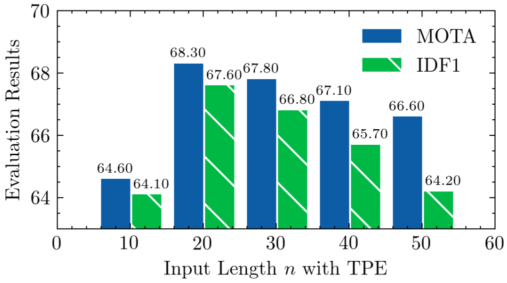

Figure 5: Analysis on tracking performance when using different lengths of input video clips with the proposed TPE approach. The results are evaluated on validation sets.

a target-guided motion model. We use motion map to extract attribute *motion* for 1) estimating motion displacements of targets, 2) generating the search region (Sec. 3.6). Without the motion map, our overall regression framework only obtains the presence and the appearance map modules at training phase, and we link trajectories without any motion prediction. As in Table 2, the highest ID Sw. indicates the association performance is imperfect during occlusions and severe motion variations. We also ablate the attribute *exit* from motion map to investigate the impact on track termination management. The quantitative results prove that introducing *exit* to our TM is benefit to further reduce FP, FN and ID Sw..

4.4 Analyses

We further compare our motion map module and association strategy with other popular alternatives to demonstrate the superiority of our proposed framework. Followed by these, we provide detailed analyses on hyper-parameters in our proposed TPE method.

Motion Model. For the study we replace our motion map module with pixel-wise optical flows from FlowNet2[19] for motion displacement estimation. As shown in Table 3, replacing the proposed motion map module with optical flow decreased the MOTA and IDF1 scores slightly by 0.4 and 0.2 points respectively. This indicates that the learned target-guided motion map is effective at estimating motion dynamics and reducing association errors.

Association Strategy. As described in Section 3.6, the extracted trajectory attributes from output TMs can help to reduce the trajectory generation to a simple nearest neighbor search. We are also interested to see if applying global assignment could further improve the tracking performance. As listed Table 3, the benefit of our simpler greedy matching is mainly the lower False Negatives because less trajectories got terminated incorrectly. But on the other hand, global assignment with Hungarian algorithm [23] did further reduce the ID Sw., at the cost of a slightly more computation.

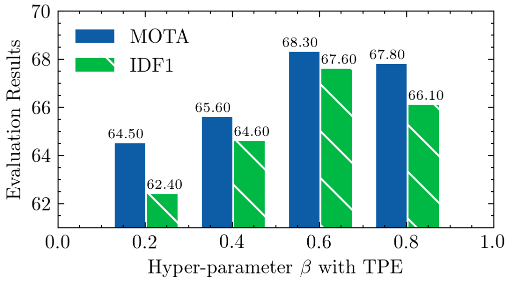

Figure 6: Analysis on tracking performance when using different hyper-parameter β with the proposed TPE approach. The results are evaluated on validation sets.

Hyper-parameters. In our framework, two hyper-parameters in the TPE process, i.e., the input length n of video clips in Eq. (7) and the parameter β in Eq. (6) need to be pre-defined. Fig. 5 demonstrates the validation results under different length n settings. Increasing n means more temporal information are engaged for extracting the trajectory attributes, and the best performance we obtained is when $n = 20$. It is observed that longer n setting may occasionally link a dead target to an incorrect but visually similar target, and thus decreased the scores. Fig. 6 illustrates the analysis on hyper-parameter β. As defined in Eq. (6), increasing β results in generating less positive target-wise distributions on TMs, while decreasing β results in bringing more positives on the contrary. Introducing more correct positive distributions helps in reducing the FNs, while introducing wrong ones will increase the FPs. We obtained the best performance when β is set to 0.6 in our experiments.

4.5 Tracking Performance

Benchmark Evaluation. Table 4 lists the evaluation results on MOT16/17/20 benchmarks in comparison with existing SOTA methods after peer reviewed. Our method outperforms SOTA methods in all three benchmarks in terms of main evaluation metrics MOTA and IDF1 scores, as well as MT, FN, ID Sw. and Frag.

Compared to the latest one-stage trackers such as Chained-Tracker [32] and CenterTrack [54], our tracker achieved higher MOTA and almost the best IDF1, which indicates our TM regression framework with the proposed TPE module for one-stage learning to track is superior than extending image-based detectors to trackers. Moreover, adopting *presence*/*enter*/*exit* as attributes gives a complete trajectory state space and results in higher MT and lower FN, which leverages the power of incorporating temporal dynamics for directly estimating and locating tracked targets instead of objects. Lower ID Sw. and Frag indicate that 1) embedding motion dynamics into a spatial-temporal target-guided motion map leads to a rather robust motion estimator, 2) introducing occlusion-aware radius helps maintain identities against occlusions.

Compared to the video-based trackers such as TubeTK [31], our tracker is better at reducing association errors against long-term occlusions and thus yields a better tracking performance. Besides, our tracker runs about ten times faster than [31].

The lower ID Sw. and Frag from graph-based tracker MLT [53] on MOT20 is mainly because its retrieved positive signals are much less than ours, which can be observed from its high FN and low

Table 4: Evaluation results with our complete setting under the private protocol. The best results are highlighted in bold. Measurements with ↑/↓ means that higher/lower score denote better performance respectively. Mode $Two-S.$, $One-S.$, $One-S.\&Vid.$ indicate two-stage, one-stage and one-stage video-based methods respectively.

Benchmark	Method	Mode	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓	Frag ↓
MOT16	DeepSORT [48]	$Two-S.$	61.4	62.2	32.8%	18.2%	12.852	56,668	781	2,008
	VMaxx [45]	$Two-S.$	62.6	49.2	32.7%	21.1%	10,604	56,182	1,389	1,534
	LM_CNN [2]	$Two-S.$	67.4	61.2	38.2%	19.2%	10,109	48,435	931	1,034
	Tube_TK_POI [31]	$One-S.\&Vid.$	66.9	62.2	39.0%	**16.1%**	11,544	47,502	1,236	1,444
	CTrackerV1 [32]	$One-S.$	67.6	57.2	32.9%	23.1%	**8,934**	48,305	1,897	3,112
	KDNT [52]	$Two-S.$	68.2	60.0	41.0%	19.0%	11,479	45,605	933	1,093
	Ours	$One-S.\&Vid.$	**70.2**	**65.4**	**42.2%**	22.3%	9,316	**44,248**	**716**	**969**
MOT17	Tube_TK [31]	$One-S.\&Vid.$	63.0	58.6	31.2%	**19.9%**	27,060	177,483	4,137	5,727
	CTrackerV1 [32]	$One-S.$	66.6	57.4	32.2%	24.2%	22,284	160,491	5,529	9,114
	CTTrack17 [54]	$One-S.\&Vid.$	67.8	**64.7**	34.6%	24.6%	**18,498**	160,332	3,039	6,102
	Ours	$One-S.\&Vid.$	**68.8**	64.6	**40.4%**	23.7%	25,722	**147,855**	**2,190**	**2,916**
MOT20	MLT [53]	$Two-S.$	48.9	54.6	30.9%	22.1%	45,660	216,803	**2,187**	**3,067**
	Ours	$One-S.\&Vid.$	**64.3**	**66.6**	**50.4%**	**14.0%**	**40,780**	**140,565**	3,379	7,405

Table 5: Run-time (ms/frame) comparison with two-stage trackers using Faster R-CNN [36] detectors.

Method	Detection ↓	Tracking ↓	Total Time ↓	FPS ↑
RAR16wVGG [9]	> 100	625	> 725	< 1.5
CNNMTT [27]	> 100	89	> 189	<5.5
POI [52]	> 60	101	> 161	<6.2
Deep SORT [48]	> 100	57	> 157	<6.5
TAP [56]	> 100	55	> 155	<7
Ours (complete)	**0**	**30**	**30**	**33**

Table 6: Tracking performance comparison with SOTA one-stage trackers on MOT16 benchmark in terms of efficiency and accuracy.

Protocol	Method	FPS ↑	MOTA ↑	IDF1 ↑
Public	Tracktor++ [3]	1.5	56.3	55.1
	RetinaTrack [26]	14.2	56.7	-
Private	Tube_TK [31]	3.0	66.9	62.2
	CenterTrack [54]	17.5	69.6	60.7
	JDE^{1088} [47]	22.2	64.4	55.8
	JDE^{864} [47]	30.3	62.1	56.9
	Ours	**33**	**70.2**	**65.4**

MOTA scores. Our tracker instead, is able to maintain promising tracking accuracy and efficiency in dense crowded scenes. We also conducted evaluations under the public protocol to compare with other popular trackers, which are included in supplementary materials along with more qualitative comparisons.

Run Time Efficiency. Table 5 investigates the run time efficiency of our proposed one-stage framework, in comparison with several two-stage online trackers [9, 27, 48, 52, 56]. It is observed that, since our tracker generates trajectories for all activated targets in one feed-forward propagation, it not only completely saved the detection time, but also required less time for tracking than most of the online trackers such as the well-known method Deep SORT [48] using Kalman Filter and re-id features for association.

We also compare the tracking performance with SOTA one-stage trackers to demonstrate the superiority of our proposed MOT framework. As shown in Table 6, Tracktor++ [3] is a strong baseline method extended from two-stage detector Faster R-CNN [36], while latest methods [26, 47, 54] adopted one-stage detectors as backbone networks including RetinaNet [25], YOLOv3 [34] and CenterNet [55] out of time-efficiency. Compared to them, our complete system does no engage any specific detection backbone and ran at $30ms/frame$ (33 FPS), faster than real-time.

5 CONCLUSIONS

We present a novel feature representation scheme and video-based framework for real-time Multiple Object Tracking by Trajectory Map regression. Different from extending image-based object detectors, we propose to learn a spatial-temporal feature representation by encoding multiple essential attributes into a continuous Trajectory Map, and conduct target locating, attributes encoding and state estimation of trajectories simultaneously using a multi-task learning network. With the proposed Temporal Priors Embedding approach and occlusion-aware radius, the complete model is capable of describing the dynamic features of long-term tracklets and can handle the entering/exiting/occlusion of targets robustly. The total framework is simple and runs rather fast at 33 FPS. We report extensive experimental analyses on several benchmarks to show that our tracker outperforms State-of-the-Art both in accuracy and efficiency. Our extensions to extract other attributes in a more complete visual tracking state space, such as *orientation*, *depth*, *interactive*, *dense*, etc., is on-going.

6 ACKNOWLEDGMENTS

This work is jointly supported by the National Key Research and Development Program of China under Grant No. 2017YFA0700800, the General Program of China Postdoctoral Science Foundation under Grant No. 2020M683490, and the Youth program of Shaanxi Natural Science Foundation under Grant No. 2021JQ-054.

REFERENCES

[1] 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*. IEEE, 1926–1933.

[2] Maryam Babaee, Zimu Li, and Gerhard Rigoll. 2019. A dual CNN–RNN for multiple people tracking. *Neurocomputing* 368 (2019), 69–83.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. 2019. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision*. 941–951.

[4] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing* 2008 (2008), 1–10.

[5] Bo Wu and R. Nevatia. 2006. Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 1. 951–958. https://doi.org/10.1109/CVPR.2006.312

[6] Afshin Dehghan, Yicong Tian, Philip HS Torr, and Mubarak Shah. 2015. Target identity-aware network flow for online multiple target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1146–1154.

[7] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. 2020. MOT20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003 [cs.CV]

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[9] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. 2018. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 466–475.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. 2017. Detect to Track and Track to Detect. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 3057–3065. https://doi.org/10.1109/ICCV.2017.330

[11] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. 2009. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence* 32, 9 (2009), 1627–1645.

[12] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. 1983. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering* 8, 3 (1983), 173–184.

[13] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 2014. 3D Traffic Scene Understanding From Movable Platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 5 (2014), 1012–1025. https://doi.org/10.1109/TPAMI.2013.185

[14] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. 2015. Joint probabilistic data association revisited. In *Proceedings of the IEEE international conference on computer vision*. 3047–3055.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[16] K. He, X. Zhang, S. Ren, and J. 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

[17] David Held, Sebastian Thrun, and Silvio Savarese. 2016. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*. Springer, 749–765.

[18] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. 2017. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3203–3212.

[19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2462–2470.

[20] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Object Detection from Video Tubelets with Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 2016). https://doi.org/10.1109/cvpr.2016.95

[21] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. 2015. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*. 4696–4704.

[22] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[23] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[24] Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 734–750.

[25] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. 2017. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2999–3007. https://doi.org/10.1109/ICCV.2017.324

[26] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. 2020. RetinaTrack: Online Single Stage Joint Detection and Tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[27] Nima Mahmoudi, Seyed Mohammad Ahadi, and Mohammad Rahmati. 2019. Multi-target tracking using CNN-based features: CNNMTT. *Multimedia Tools and Applications* 78, 6 (2019), 7077–7096.

[28] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* (2016).

[29] Anton Milan, Stefan Roth, and Konrad Schindler. 2013. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence* 36, 1 (2013), 58–72.

[30] Anton Milan, Konrad Schindler, and Stefan Roth. 2013. Detection-and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3682–3689.

[31] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. 2020. TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. 2020. Chained-Tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-Object Detection and Tracking. In *Computer Vision – ECCV 2020*. Springer International Publishing, Cham, 145–161.

[33] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. IEEE, 1201–1208.

[34] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[35] Donald Reid. 1979. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control* 24, 6 (1979), 843–854.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[37] 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 *European Conference on Computer Vision*. Springer, 17–35.

[38] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. 2016. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*. Springer, 84–99.

[39] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. 2018. CrowdHuman: A Benchmark for Detecting Human in a Crowd. arXiv:1805.00123 [cs.CV]

[40] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[41] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. 2017. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5620–5629.

[42] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*. Springer, 100–111.

[43] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. 2017. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3539–3548.

[44] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. 2019. MOTS: Multi-object tracking and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7942–7951.

[45] Xingyu Wan, Jinjun Wang, Zhifeng Kong, Qing Zhao, and Shunming Deng. 2018. Multi-Object Tracking Using Online Metric Learning with Long Short-Term Memory. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 788–792.

[46] Xinchao Wang, Engin Türetken, Francois Fleuret, and Pascal Fua. 2015. Tracking interacting objects using intertwined flows. *IEEE transactions on pattern analysis and machine intelligence* 38, 11 (2015), 2312–2326.

[47] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. 2020. Towards Real-Time Multi-Object Tracking. In *Computer Vision – ECCV 2020*. Springer International Publishing, Cham, 107–122.

[48] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 3645–3649.

[49] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5987–5995. https://doi.org/10.1109/CVPR.2017.634

[50] Bo Yang and Ram Nevatia. 2012. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1918–1925.

[51] Fan Yang, Wongun Choi, and Yuanqing Lin. 2016. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2129–2137.

[52] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. 2016. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*. Springer, 36–42.

[53] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong. 2020. Multiplex Labeling Graph for Near-Online Tracking in Crowded Scenes. *IEEE Internet of Things Journal* 7, 9 (2020), 7892–7902. https://doi.org/10.1109/JIOT.2020.2996609

[54] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2020. Tracking Objects as Points. In *Computer Vision – ECCV 2020*. Springer International Publishing, Cham, 474–490.

[55] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850* (2019).

[56] Zongwei Zhou, Junliang Xing, Mengdan Zhang, and Weiming Hu. 2018. Online multi-target tracking with tensor-based high-order graph matching. In *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 1809–1814.