SiamMOT: Siamese Multi-Object Tracking

Bing Shuai   Andrew Berneshawi   Xinyu Li   Davide Modolo   Joseph Tighe
Amazon Web Services (AWS)
{bshuai, bernea, xxnl, dmodolo, tighej}@amazon.com

Abstract

In this paper, we focus on improving online multi-object tracking (MOT). In particular, we introduce a region-based Siamese Multi-Object Tracking network, which we name SiamMOT. SiamMOT includes a motion model that estimates the instance's movement between two frames such that detected instances are associated. To explore how the motion modelling affects its tracking capability, we present two variants of Siamese tracker, one that implicitly models motion and one that models it explicitly. We carry out extensive quantitative experiments on three different MOT datasets: MOT17, TAO-person and Caltech Roadside Pedestrians, showing the importance of motion modelling for MOT and the ability of SiamMOT to substantially outperform the state-of-the-art. Finally, SiamMOT also outperforms the winners of ACM MM'20 HiEve Grand Challenge on HiEve dataset. Moreover, SiamMOT is efficient, and it runs at 17 FPS for 720P videos on a single modern GPU.

1. Introduction

Multi-object tracking (MOT) is the problem of detecting object instances and then temporally associating them to form trajectories. Early works [1, 3, 4, 17, 29, 33, 46, 49, 53, 53, 59, 64, 65, 71, 72] formulate instance association as a graph-based optimization problem under the "tracking-by-detection" paradigm, in which a node represents a detection and an edge encodes the likelihood of two nodes being linked. In practice, they use a combination of visual and motion cues to represent each node, which often requires expensive computation. Furthermore, they usually construct a large offline graph, which is non-trivial to solve, making them inapplicable for real-time tracking. Recently, online trackers [5,7,60,76] started to emerge, as they are more desirable in real-time tracking scenarios. They focus on improving local linking over consecutive frames rather than building an offline graph to re-identify instances across large temporal gaps. Among these, some recent works [5, 75] have pushed *online* MOT into state-of-the-art territory, making them very competitive.

In this work, we explore the importance of modelling motion in *online MOT* by building upon "Simple and Online Realtime Tracking" (SORT) [7,60] that underlies recent state-of-the-art models [5, 76]. In SORT, a better motion model is the key to improving its local linking accuracy. For example, SORT [7] uses Kalman Filters [31] to model the instance's motion with simple geometric features, while the more recent state-of-the-art trackers [5, 76] learn a deep network to predict the displacement (motion) of instances based on both visual and geometric features, significantly outperforming the simpler SORT.

We conduct our motion modelling exploration by leveraging a region-based Siamese Multi-Object Tracking network, which we name **SiamMOT**. We combine a region-based detection network (Faster-RCNN [45]) with two motion models inspired by the literature on Siamese-based single-object tracking [6, 18, 22, 35, 36]: an implicit motion model (IMM) and an explicit motion model (EMM). Differently from CenterTrack [76] that implicitly infers the motion of instances with point-based features [16, 44, 56], SiamMOT uses region-based features and develops (explicit) template matching to estimate instance motion, which is more robust to challenging tracking scenarios, such as fast motion.

We present extensive ablation analysis on three different multi-person tracking datasets. Our results suggest that instance-level motion modelling is of great importance for robust online MOT, especially in more challenging tracking scenarios. Furthermore, we show that the motion models of SiamMOT can improve tracking performance substantially, especially when cameras are moving fast and when people's poses are deforming significantly. On the popular MOT17 Challenge [42] SiamMOT with EMM achieves **65.9** MOTA / **63.3** IDF1 with a DLA-34 [69] backbone by using *public* detection, outperforming all previous methods. Moreover, on the recently introduced large-scale TAO-person dataset [14], SiamMOT substantially improves over the state-of-the-art Tracktor++ [5] from **36.7** to **41.1** TrackAP [14, 66]. Finally, we benchmark SiamMOT on the Human In Events (HiEve) dataset [41], where it outperforms the winner of the ACM MM'20 grand HiEve challenge [40].

2. Related work

2.1. Siamese trackers in SOT

Single object tracking (SOT) refers to tracking a given object of interest, which is usually specified in the first frame and could belong to any semantic object class. Instead of detecting pre-defined objects in a frame and linking them back to earlier tracked instances, single object trackers (SOT) usually model the motion of the object of interest directly to predict its trajectory. Siamese-based trackers [6, 18, 22, 23, 26, 28, 35, 36, 54, 57, 73, 74, 78] are a family of state-of-the-art SOT. As the name suggests, Siamese trackers operate on pairs of frames. Their goal is to track (by matching) the target object in the first frame within a search region in the second frame. This matching function is usually learned offline on large-scale video and image datasets.

In this paper, we formulate Siamese trackers within an end-to-end trainable multi-object tracking network (SiamMOT). The closest work to ours is DeepMOT that also trains Siamese trackers with other components under the proposed MOT training framework. However, Deep-MOT focuses on improving the structured loss in MOT rather than formulating the detector and tracker in a unified network, so an off-the-shelf single object tracker is needed in DeepMOT. Finally, while we take inspiration from particular Siamese trackers [22, 34, 36], our formulation is generic enough that other Siamese trackers can easily be adapted in our MOT framework.

Siamese network. It's worth noting that Siamese trackers are different from general Siamese networks [34, 52, 58]. Siamese networks usually learn a affinity function between two detected instances, whereas Siamese trackers learn a matching function that is used to search for a detected instance within a larger contextual region.

2.2. Tracking-by-Detection in MOT

Many works tackle multi-object tracking (MOT) by adopting the "tracking-by-detection" paradigm [1, 3, 4, 11, 17, 19, 29, 33, 34, 46, 47, 49, 53, 59, 64, 65, 71, 72], where objects instances are first detected in each frame and then associated across time based on their visual coherence and spatial-temporal consistency. Some of these works focused on learning new functions to evaluate short-term associations more robustly [10, 19, 34, 46, 47, 49, 53, 53, 64, 72, 72]. Others, instead, focused on learning how to output more temporally consistent long-term tracks by optimizing locally connected graphs [1, 3, 4, 17, 29, 33, 46, 49, 53, 53, 59, 64, 65, 71, 72]. Many of these approaches are inefficient, as they employ separate computationally expensive cues, like object detection [12, 20, 27, 45], optical flow [10, 15, 51, 53], and re-identification [30, 46, 53, 53, 75].

Online MOT. Online MOT refers to performing instance association on the fly without knowledge of future frames [2, 5, 7, 32, 60, 63, 76]. Therefore, online MOT focuses more on accurate local association rather than global-optimal association in which detections can be linked across long temporal gaps (as in offline graph modelling). It has seen a resurgence of popularity recently as new models are efficient enough to be applicable to real-time tracking. For example, Ban et al. [2] formulated it in a probabilistic framework by using a variational expectation maximization algorithm to find the tracks. Xiang et al. [63] used Markov Decision Processes and reinforcement learning for online instance association. Bewley et al. [7, 60] developed simple object and realtime tracking (SORT) for quick online instance association. SORT has been widely used in recent deep neural network based models [5, 76] which achieve state-of-the-art performance on public MOT datasets. Our SiamMOT is based on SORT, and we explore how to improve its tracking performance.

Motion modelling in SORT. The original SORT [7] only used geometric features of tracks (location, box shape, etc) in its motion model to track locations across frames. Later, Wojke et al. [60] improved SORT by incorporating visual features into the motion model to link the detected instances. Recently, Bergmann et al. [5] and Zhou et al. [76] jointly learned the motion model with the detector such that both visual and geometric features are used. In detail, Tracktor [5] leveraged a two stage detector [45] to regress from previous person's location to current frame; Center-Track [76] adopted a track branch to regress the displacement of object centers between frames. In this paper, we explore how to improve the motion model in SORT-based tracking model – SiamMOT, and more importantly how it leads to improved MOT accuracy.

3. SiamMOT: Siamese Multi-Object Tracking

SiamMOT builds upon Faster-RCNN object detector [20, 27, 45], which consists of a Region Proposal Network (RPN) and a region-based detection network. On top of the standard Faster-RCNN, SiamMOT adds a region-based Siamese tracker to model instance-level motion. As shown in Fig. 1, SiamMOT takes as input two frames $\mathbf{I}^t, \mathbf{I}^{t+\delta}$ together with a set of detected instances $\mathbf{R}^t = \{R_1^t, \dots R_i^t, \dots\}$ at time t. In SiamMOT, the detection network outputs a set of detected instances $\mathbf{R}^{t+\delta}$, while the tracker propagates $\mathbf{R^t}$ to time $t + \delta$ to generate $\tilde{\mathbf{R}}^{t+\delta}$.

As in SORT, SiamMOT contains a motion model that *tracks* each detected instance from time t to $t + \delta$ by propagating the bounding box R_i^t at time t to $\tilde{R}_i^{t+\delta}$ at $t + \delta$; and a spatial matching process that *associates* the output of tracker $\tilde{R}_i^{t+\delta}$ with the detections $R_i^{t+\delta}$ at time $t + \delta$ such that detected instances are linked from t to $t + \delta$.
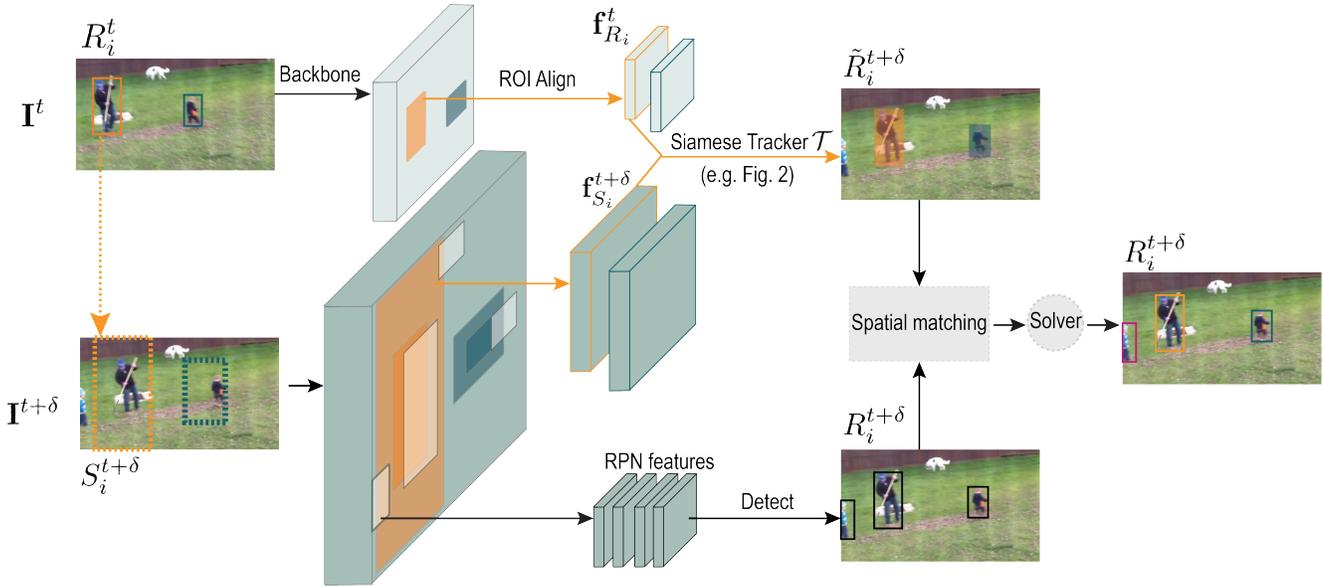
Figure 1: *(Best viewed in color) SiamMOT is a region-based multi-object tracking network that detects and associates object instances simultaneously. The Siamese tracker models the motion of instances across frames and it is used to temporally link detection in online multi-object tracking. Backbone feature map for frame $\mathbf{I}^t$ is visualized with 1/2 of its actual size.*

In the next section we introduce how our Siamese tracker models instance motion in SiamMOT (Sec. 3.1) and present two variants of Siamese trackers in Sec. 3.2 and Sec. 3.3. Finally, we provide the details for training and inference (Sec. 3.4).

3.1. Motion modelling with Siamese tracker

In SiamMOT, given a detected instance i at time t, the Siamese tracker searches for that particular instance at frame $\mathbf{I}^{t+\delta}$ in a contextual window around its location at frame $\mathbf{I}^t$ (i.e, R_i^t). Formally,

$$(v_i^{t+\delta}, \tilde{R}_i^{t+\delta}) = \mathcal{T}(\mathbf{f}_{R_i}^t, \mathbf{f}_{S_i}^{t+\delta}; \Theta) \tag{1}$$

where $\mathcal{T}$ is the learnable Siamese tracker with parameters Θ, $\mathbf{f}_{R_i}^t$ is the feature map extracted over region R_i^t in frame $\mathbf{I}^t$, and $\mathbf{f}_{S_i}^{t+\delta}$ is the feature map extracted over the search region $S_i^{t+\delta}$ in frame $\mathbf{I}^{t+\delta}$. We compute $S_i^{t+\delta}$ by expanding R_i^t by a factor r (> 1) while maintaining the same geometric center (e.g., dashed bounding box in Fig. 1). We extract features $\mathbf{f}_{R_i}^t$ and $\mathbf{f}_{S_i}^{t+\delta}$ using the region of interest align (ROIAlign) layer of Mask-RCNN [27]. Finally, $v_i^{t+\delta}$ is the confidence of visibility for detected instance i at time $t + \delta$. As long as the instance is visible in $S_i^{t+\delta}$, $\mathcal{T}$ should produce a high score $v_i^{t+\delta}$, otherwise $\mathcal{T}$ should produce a low score. Note how this formulation is reminiscent of that of Siamese-based single-object trackers [6, 28, 35, 36] and specifically, how they model the instance's motion between frames.

In the context of multi-object tracking, we apply Eq. 1 multiple times, once for each detected instance $R_i^t \in \mathbf{R}^t$. Importantly, our SiamMOT architecture allows these operations to run in parallel and only requires the backbone features to be computed once, making online tracking inference efficient.

We conjecture that motion modelling is particularly important for online MOT. Specifically, association between R^t and $R^{t+\delta}$ will fail if 1) $\tilde{R}^{t+\delta}$ does not match to the right instance in $R^{t+\delta}$ or 2) $v_i^{t+\delta}$ is low for a visible person at $t + \delta$. Previous works [5, 76] approach the problem of regressing $\tilde{R}^{t+\delta}$ from the previous location (i.e. R_i^t) by feeding the model with features from both frames. By doing so these works aim to implicitly model the instance's motion in the network. However, as research in single-object tracking [6, 22, 35, 36] reveals, finer-grained spatial-level supervision is of great significance to explicitly learn a robust target matching function in challenging scenarios. Based on this rationale, we present two different parameterizations of $\mathcal{T}$ in SiamMOT – an implct motion model in Sec. 3.2 and an explicit motion model in Sec. 3.3.

3.2. Implicit motion model

Implicit motion model (IMM) uses an MLP to implicitly estimate the instance-level motion between two frames. In detail, the model concatenates $\mathbf{f}_{S_i}^t$ and $\mathbf{f}_{S_i}^{t+\delta}$ and feeds that to an MLP that predicts the visibility confidence v_i and the
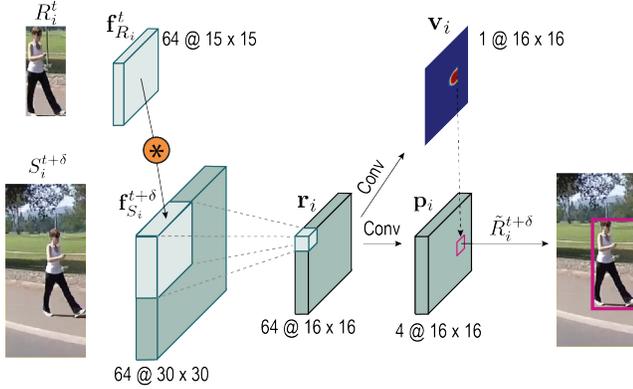
Figure 2: *Network architecture of Explicit Motion Model (EMM),* $*$ *represents channel-wise cross correlation operator.*

relative location and scale changes:

$$m_i = [\frac{x_i^{t+\delta} - x_i^t}{w_i^t}, \ \frac{y_i^{t+\delta} - y_i^t}{h_i^t}, \log\frac{w_i^{t+\delta}}{w_i^t} \ \log\frac{h_i^{t+\delta}}{h_i^t}] \quad (2)$$

in which $(x_i^t, y_i^t, w_i^t, h_i^t)$ is the parameterization of R_i^t. We can trivially derive $\tilde{R}^{t+\delta}$ from an inversion transformation of Eq. 2 by taking as input R_i^t and m_i.

Loss. Given a triplet $(R_i^t, S_i^{t+\delta}, R_i^{t+\delta})$, we train IMM with the following training loss:

$$\mathbf{L} = \ell_{focal}(v_i, v_i^*) + \mathbb{1}[v_i^*]\,\ell_{reg}(m_i, m_i^*) \quad (3)$$

where v_i^* and m_i^* refer to ground truth values derived from $R_i^{t+\delta}$, $\mathbb{1}$ is the indicator function, ℓ_{focal} the focal loss for classification [38] and ℓ_{reg} the commonly used smooth ℓ_1 loss for regression. Please refer to the supplementary material for the network architecture.

3.3. Explicit motion model

Inspired by the literature on single-object tracking [6,22, 35, 36, 54], we propose an explicit motion model (EMM, Fig.2) in SiamMOT. Specifically, it uses a channel-wise cross-correlation operator ($*$) to generate a pixel-level response map $\mathbf{r}_i$, which has shown to be effective in modelling dense optical flow estimation [15] and in SOT for instance-level motion estimation [6, 22, 35, 36]. In SiamMOT, this operation correlates each location of the search feature map $\mathbf{f}_{S_i}^{t+\delta}$ with the target feature map $\mathbf{f}_{R_i}^t$ to produce $\mathbf{r}_i = \mathbf{f}_{S_i}^{t+\delta} * \mathbf{f}_{R_i}^t$, so each map $\mathbf{r}_i[k,:,:]$ captures a different aspect of similarity. Inspired by FCOS [56], EMM uses a fully convolutional network ψ to detect the matched instances in $\mathbf{r}_i$. Specifically, ψ predicts a dense visibility confidence map $\mathbf{v}_i$ indicating the likelihood of each pixel to contain the target object, and a dense location map $\mathbf{p}_i$

that encodes the offset from that location to the top-left and bottom-right bounding box corners. Thus, we can derive the instance region at (x, y) by the following transformation $\mathcal{R}(\mathbf{p}(x, y)) = [x - l, y - t, x + r, y + b]$ in which $\mathbf{p}(x, y) = [l, t, r, b]$ (the top-left and bottom-right corner offsets). Finally, we decode the maps as follows:

$$\tilde{R}_i^{t+\delta} = \mathcal{R}(\mathbf{p}_i(x^*, y^*)); \quad v_i^{t+\delta} = \mathbf{v}_i(x^*, y^*)$$
$$\mathbf{s.t.}(x^*, y^*) = \underset{x,y}{\operatorname{argmax}}(\mathbf{v}_i \odot \boldsymbol{\eta}_i) \quad (4)$$

where $\odot$ is the element-wise multiplication, $\boldsymbol{\eta}_i$ is a penalty map that specifies a non-negative penalty score for the corresponding candidate region as follows:

$$\boldsymbol{\eta}_i(x, y) = \lambda\mathcal{C} + (1 - \lambda)\mathcal{S}(\mathcal{R}(\mathbf{p}(x, y)), R_i^t) \quad (5)$$

where λ is a weighting scalar ($0 \le \lambda \le 1$), $\mathcal{C}$ is the cosine-window function w.r.t the geometric center of the previous target region $\mathcal{R}_i^t$ and $\mathcal{S}$ is a Guassian function w.r.t the relative scale (height / width) changes between the candidate region $(\mathbf{p}(x, y))$ and R_i^t. The penalty map $\boldsymbol{\eta}_i$ is introduced to discourage dramatic movements during the course of tracking, similar to that in [18,22,35,36].

Loss. Given a triplet $(R_i^t, S_i^{t+\delta}, R_i^{t+\delta})$, we formulate the training loss of EMM as follows:

$$\mathbf{L} = \sum_{x,y} \ell_{focal}(\mathbf{v}_i(x, y), \mathbf{v}_i^*(x, y))$$
$$+ \sum_{x,y} \mathbb{1}[\mathbf{v}_i^*(x, y) = 1](w(x, y) \cdot \ell_{reg}(\mathbf{p}_i(x, y), \mathbf{p}_i^*(x, y)))$$
$$(6)$$

where (x, y) enumerates all the valid locations in $S_i^{t+\delta}$, ℓ_{reg} is the IOU Loss for regression [13,70] and ℓ_{focal} is the focal loss for classification [38]. Finally, $\mathbf{v}_i^*$ and $\mathbf{p}_i^*$ are the pixel-wise ground truth maps. $\mathbf{v}_i^*(x, y) = 1$ if (x, y) is within $R_i^{*t+\delta}$ and 0 otherwise. $\mathbf{p}_i^*(x, y) = [x - x_0^*, y - y_0^*, x_1^* - x, y_1^* - y]$ in which (x_0^*, y_0^*) and (x_1^*, y_1^*) corresponds to the coordinates of the top-left and the bottom-right corner of ground truth bounding box $R_i^{t+\delta}$. Similar to [77], we modulate ℓ_{reg} with $w(x, y)$, which is the centerness of location (x, y) w.r.t to the target instance $R_i^{t+\delta}$ and is defined as $w(x, y) = \sqrt{\frac{\min(x-x_0, x_1-x)}{\max(x-x_0, x_1-x)} \cdot \frac{\min(y-y_0, y_1-y)}{\max(y-y_0, y_1-y)}}$.

EMM improves upon the IMM design in two ways. First it uses the channel independent correlation operation to allow the network to explicitly learn a matching function between the same instance in sequential frames. Second, it enables a mechanism for finer-grained pixel-level supervision which is important to reduce the cases of falsely matching to distractors.

3.4. Training and Inference

We train SiamMOT in an end-to-end fashion with the following loss $\ell = \ell_{rpn} + \ell_{detect} + \ell_{motion}$, in which

ℓ_{rpn} and ℓ_{detect} are the standard losses for RPN [45] and the detection sub-network [20] in Faster-RCNN. $\ell_{motion} = \sum_{x_i \in \mathcal{X}} \mathbf{L}(x_i)$ is used to train the Siamese tracker, wherein $\mathcal{X} = \cup_{i=1}^{M} (R_i^t, S_i^{t+\delta}, R_i^{t+\delta})$ are training triplets. Note that $R_i^{t+\delta} = \varnothing$ if R_i^t does not include a ground truth instance or the instance in R_i^t is not visible in $S_i^{t+\delta}$. Similar to Faster-RCNN training, we sample R_i^t from the outputs of the RPN [45].

At inference, a standard IOU-based NMS operation is first used on the outputs of the detection sub-network ($R^{t+\delta}$ in Fig. 1) and on those of the Siamese tracker ($\tilde{R}^{t+\delta}$ in Fig. 1) independently. Next, the following spatial matching process is used to merge $R^{t+\delta}$ and $\tilde{R}^{t+\delta}$: detections that spatially match ($IOU \geq 0.5$) to any tracked instance are suppressed and thus removed. Then, we adopt a standard *online* solver as that in [5, 7, 60, 76]: 1) a trajectory is continued if its visibility confidence (v_i^t) is above α; 2) a trajectory is born if there is a non-matched detection and its confidence is above β and 3) a trajectory is killed if its visibility confidence (v_i^t) is below α for consecutive τ frames.

Short occlusion handling. In the case of short occlusions, the visibility confidence for the target would be low (lower than the threshold α). Instead of killing them, we keep those tracks in memory and continue searching for them in future frames (up to $\tau > 1$ frames) to check whether they can be re-instated. We use the last predicted location and its corresponding feature as the searching template.

4. Experimental settings

4.1. Datasets and Metrics

MOT17 [42] is the most widely used multi-person tracking benchmark. It consists of 7 training and 7 test videos, ranging from from 7 to 90 seconds long. The videos feature crowded scenes in indoor shopping malls or outdoor streets. We follow the evaluation protocol of [42] and report our results using several metrics: MOTA (Multiple Object Tracking Accuracy), IDF1 (ID F1 score), FP (False Positives), FN (False Negatives) and IDsw (ID switches).

TAO-person [14] is a newly-established large scale multi-person tracking benchmark. It is a subset of the TAO dataset [14] and it consists of 418 training and 826 validation videos. To include a large variability of scenes, the videos are collected by mixing existing datasets like AVA [21] (generic movies), Charades [50] (indoor activities), BDD [68] (streets), Argoverse [9] (streets) and other sports videos. This dataset contains rich motion artifacts (e.g. motion and defocus blur), as well as diverse person motion patterns (Fig. 3c), which makes tracking persons challenging. We follow the evaluation protocol of [14] and use the provided toolbox to report Federated Track-AP (TAP). Federated evaluation [24] is used because not all

videos are exhaustively annotated. Different from MOTA, Track-AP [66] highlights the temporal consistency of the underlying trajectories.

Caltech Roadside Pedestrians (CRP) [25] is a dataset for person analysis in videos. It consists of 7 videos, each roughly 20 minutes long. The videos are captured from a camera mounted to a car while driving, and they mainly feature outdoor roadside scenes. Due to the fast camera motion, the pedestrians appear as they are moving relatively much faster than in other datasets (Fig. 3b). We report results on the same metrics used for MOT17.

Datasets analysis. Each of these datasets contains different challenges for tracking. For example, tracking people in MOT17 is challenging due to occlusion and crowded scenes, even though people do not move fast and their poses are constant (i.e., standing). In contrast, scenes in CRP are not as crowded, but the camera motion is very large and the pedestrian's position changes quickly. Finally, TAO includes a wide range of scene types and video corruption artifacts. As we focus on modelling short term motion for tracking, here we examine the characteristics of motion in each of these datasets. Towards this, we calculate the ground truth motion vector $\mathbf{m}$ as in Eq. 3 for every person, between two consecutive annotated frames. As videos are not annotated densely (i.e., every frame), we normalize $\mathbf{m}$ by δ (their time difference). We present dataset-specific histograms in Fig. 3. People in MOT17 dataset have relatively small motion compared to those in TAO and CRP.

4.2. Implementation details

Network. We use a standard DLA-34 [69] with feature pyramid [37] as the Faster-RCNN backbone. We set $r = 2$, so that our search region is $2\times$ the size of the tracking target. In IMM, $\mathbf{f}_{S_i}^t$ and $\mathbf{f}_{S_i}^{t+\delta}$ have the same shape $\mathbb{R}^{c \times 15 \times 15}$ and the model is parametrized as a 2-layer MLP with 512 hidden neurons. In EMM, instead, $\mathbf{f}_{R_i}^t \in \mathbb{R}^{c \times 15 \times 15}$ and $\mathbf{f}_{S_i}^{t+\delta} \in \mathbb{R}^{c \times 30 \times 30}$, so that they are at the same spatial scale; the model is a 2-layer fully convolutional network, with stacks of 3×3 convolution kernels and group normalization [62]).

Training samples. As previously mentioned, we train SiamMOT on pairs of images. When video annotations are not available, we follow [28, 76] by employing *image training*, in which spatial transformation (crop and re-scale) and video-mimicked transformation (motion blur) are applied to an image such that a corresponding image pair is generated. When video annotations are available, we use *video training*, in which we sample pairs of two random frames that are at most 1 second apart.

Training. We jointly train the tracker and detection network. We sample 256 image regions from the output of the RPN to train them. We use SGD with momentum as
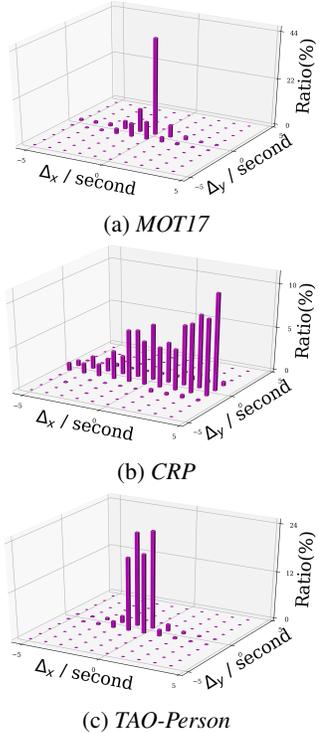
(a) *MOT17*

(b) *CRP*

(c) *TAO-Person*

Figure 3: *3D histogram of normalized motion offset per second across different datasets.*

the optimizer, and we train our model for $25K$ and $50K$ iterations, for CrowdHumand [48] and COCO [39] datasets respectively. We resize the image pair during training such that its shorter side has 800 pixels. We start training with a learning rate of 0.02 and decrease it by factor 10 after 60% of the iterations, and again after 80%. We use a fixed weight decay of 10^{-4} and a batch size of 16 image pairs.

Inference. We empirically set linking confidence $\alpha = 0.4$ and detection confidence $\beta = 0.6$, and we present the sensitivity analysis of α and β in the supplementary material. We keep a trajectory active until it is unseen for $\tau = 30$ frames.

5. Ablation analysis

We carry out ablation analysis on MOT17, CRP and TAO-person, which are considerably different from each other (sec. 4.1, Fig. 3) and provide a good set for ablation study. We adopt *image training* to train SiamMOT, as we don't have large-scale video annotations to train a generalized model. Specifically, we train models from the full-body annotation of CrowdHuman [48], and evaluate it on MOT17-train and CRP datasets as they have amodal bounding box annotation. We train models from visible-body annotations from CrowdHuman and COCO [39] and evaluate it on the TAO-person dataset. We do this to try to keep the

models as comparable as possible while still adhering to the annotation paradigm of each dataset (amodal vs modal person bounding boxes). In order to directly compare frame-to-frame tracking, we adopt the same solver as that in Tracktor [5], in which the trajectory is killed immediately if it is unseen (i.e. $\tau = 1$ frame).

5.1. Instance-level motion modelling

We investigate the benefits of motion modelling for MOT (Table 1). We compare SiamMOT with IMM and EMM against two baselines: (1) our implementation of Tracktor [5], which we obtain by removing the Siamese tracker from SiamMOT and instead use the detection network to regress the location of the target in the current frame, and (2) Tracktor + Flow, that adds a flow-based model to estimate the movement of people across frames. This flow-based model can be considered a simple forward tracker that "moves" the previous target region to the current frame and then uses the detection network (as in Tracktor) to regress to its exact location in the current frame. The movement of the person instance is estimated by taking the median flow field of its constituent pixels. In our experiments we use a pre-trained state-of-the-art PWC-net [51] to estimate the pixel-wise optical flow field. Finally, for fair comparison we use the same detections for all four models.

Results show that our implementation of Tracktor achieves competitive results on both MOT17 and TAO-person (higher than those reported by [14], [5]), but performs poorly on CRP, as its motion model is too weak to track people that move too fast. Adding flow to Tracktor significantly improves its performance (Tracktor + Flow), especially on the challenging CRP and TAO-person datasets. SiamMOT improves these results even further, for both IMM and EMM. The performance gap is especially interesting on the CRP dataset, where both MOTA and IDF1 increased substantially (i.e., +35 MOTA and +25 IDF1 over Tracktor + Flow). Between these, EMM performs similar to IMM on CRP, but significantly better on MOT17 and TAO-person. This shows the importance of explicit template matching, which is consistent with what observed in the SOT literature [34, 35]. Finally, note that tracking performance keeps increasing as we employ better motion models (i.e., Tracktor < Flow < IMM < EMM). This further validates the importance of instance-level motion modelling in MOT. In addition, SiamMOT are significantly more efficient than Tracktor + Flow, in which flow does not share computation with Tracktor.

5.2. Training of SiamMOT: triplets sampling

We now evaluate how the distribution of triplets used to train SiamMOT (sec. 3.4) affects its tracking performance.

Models		MOT17					Caltech Roadside Pedestrians (CRP)					TAO-person
	Runtime	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDsw ↓	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDsw ↓	TAP@0.5 ↑
Faster-RCNN (Tracktor)	23.0 fps	58.6	53.0	3195	42488	858	15.9	25.1	632	21238	1126	29.1%
Faster-RCNN + Flow	12.5 fps	60.3	54.8	3518	40387	716	41.8	56.4	2381	11934	1594	32.8%
Faster-RCNN + IMM	19.5 fps	61.5	57.5	5730	36863	678	76.8	81.2	2583	2391	1377	34.7%
Faster-RCNN + EMM	17.6 fps	63.3	58.4	5726	34833	671	76.4	81.1	2548	2575	1311	35.3%

Table 1: *Results on MOT17 train, Caltech Roadside Pedestrians and TAO-Person datasets. FPS are calculated based on MOT17 videos that are resized to 720P. IMM and EMM are the motion model presented for SiamMOT.*

Sampled triplets		MOT17				TAO-person
	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDsw ↓	TAP@0.5 ↑
P + H	59.7	58.6	9639	34976	618	34.2%
P + N	62.7	58.3	6275	34955	697	35.0%
P + H + N	63.3	58.4	5726	34833	671	35.3%

Table 2: *Effects of sampled triplets for training forward tracker in SiamMOT. P / N / H are positive / negative / hard training triplet. P+H triplets are usually used in single-object tracking.*

τ (frames)		MOT17				TAO-person
	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDsw ↓	TAP@0.5 ↑
1	63.3	58.4	5726	34833	671	35.3%
5	63.5	60.0	5818	34497	622	35.6%
15	63.4	60.8	5979	34454	616	36.3%
30	63.3	60.6	6106	34465	658	36.6%
60	63.0	60.2	6510	34385	699	37.2%

Table 3: *Results of SiamMOT inference that terminates active trajectories after they are unseen within τ consecutive frames.*

Given a set of training triplets $\mathcal{X} = \cup_{i=1}^{N}(R_i^t, S_i^{t+\delta}, R_i^{t+\delta})$ from image pair $\{\mathbf{I}^t, \mathbf{I}^{t+\delta}\}$, a triplet can be either negative, positive or hard. It is negative (N) when R_i^t does not include a person, positive (P) when R_i^t and $S_i^{t+\delta}$ includes the same person, and hard (H) when R_i^t includes a person, but $S_i^{t+\delta}$ does not include the target person.

Similar to the training of SOT, we start by training the Siamese tracker with positive and hard negative (P + H) triplets. As results in Tab. 2 shows, the model achieves reasonable IDF1 on MOT17, which means that the tracker can follow a true person quite robustly, but it achieves relatively low MOTA, as it occasionally fails to kill false positive tracks. This is because the Siamese tracker in SiamMOT usually starts with noisy detection rather than with human-annotated regions (as in SOT). Instead, P + N performs better and combining all of them P + H + N achieves the best results overall.

5.3. Training of SiamMOT: joint training

We now investigate the importance of training the region-based detection network jointly with our Siamese tracker. First, we look at the impact that joint training has on the accuracy of our tracker and later on the accuracy of the person detector.

Tracking performance. We train a model only with the Siamese tracker (i.e. detection branch is discarded) and utilize the same detections used in the experiments presented in sec. 5.1 and Tab. 1. The MOTA achieved by EMM on MOT17 is 63.3 with joint training vs 61.5 without. This gap shows the benefits of joint training.

Detection performance. We compare two Faster-RCNN models trained with and without our Siamese tracker on MOT17. These models achieve 73.3% and 73.4%

AP@IOU=0.5, which indicates that the joint training in SiamMOT has no negative impact on the detection network.

Overall, these results show that joint training is very important for SiamMOT and leads to the best results.

5.4. Inference of SiamMOT

Finally, we investigate how the inference of SiamMOT affects MOT performance. Similar to Tracktor [5] and CenterTrack [76], SiamMOT focuses on improving local tracking as long as the person is visible. However, a person can be shortly invisible due to occlusion (e.g. when crossing each other) that is common in crowded scenes such as MOT17. In order to track through these cases, we allow SiamMOT to track forward even when the trajectory is not visible, i.e. the tracker does not terminate the trajectory until it fails to track the corresponding target for τ consecutive frames. Results in Tab. 3 show that tracking performance increases with τ, especially IDF1 score / TrackAP that measures the temporal consistency of trajectories. This means that our tracker is capable of tracking beyond few consecutive frames. Results also show that the improvement saturates around $\tau = 30$ (1s for 30 FPS videos). The reason is that people have likely moved outside of our search region by that time. In the future we will explore improving the motion modelling of the tracker in SiamMOT such that it can track through longer occlusions.

6. Comparison to State-of-the-art

Finally, we compare our SiamMOT with state-of-the-art models on three challenging multi-person tracking datasets: MOT17 [42], TAO-person [14] and HiEve Challenge [41].

Method	MOTA	IDF1	MT	ML	FP	FN	IDsw
STRN [64]	50.9	56.5	20.1%	37.0%	27532	246924	2593
Tracktor++ [5]	53.5	52.3	19.5%	36.6%	12201	248047	2072
DeepMOT [65]	53.7	53.8	19.4%	36.6%	11731	247447	1947
Tracktor++ v2 [5]	56.5	55.1	21.1%	35.3%	8866	235449	3763
NeuralSolver [8]	58.8	61.7	28.8%	33.5%	17413	213594	1185
CenterTrack [76]	61.5	59.6	26.4%	31.9%	14076	200672	2583
SiamMOT	**65.9**	**63.3**	34.6%	23.9%	18098	170955	3040

Table 4: *Results on MOT17 test set with public detection.*

Method	Backbone	TAP@0.5	TAP@0.75
Tractor [14]	ResNet-101	26.0%	n/a
Tracktor++ [14]	ResNet-101	36.7%	n/a
SiamMOT	ResNet-101	**41.1**%	**23.0**%
SiamMOT	DLA-169	42.1%	24.3%
SiamMOT+	DLA-169	**44.3**%	**26.2**%

Table 5: *Results on TAO-person validation set.*

Method	MOTA	IDF1	MT	ML	FP	FN	IDsw
DeepSORT [60]	27.1	28.6	8.5%	41.5%	5894	42668	2220
FCS-Track [40]	47.8	49.8	25.3%	30.2%	3847	30862	1658
Selective JDE [61]	50.6	56.8	25.1%	30.3%	2860	29850	1719
LinkBox [43]	51.4	47.2	29.3%	29.0%	2804	29345	1725
SiamMOT (DLA-34)	51.5	47.9	25.8%	26.1%	3509	28667	1866
SiamMOT (DLA-169)	**53.2**	51.7	26.7%	27.5%	2837	**28485**	1730

Table 6: *HiEve benchmark leaderboard (public detection).*

MOT17 (Tab. 4). We report results on the test set using publicly released detections, as done for the official MOT17 Challenge. We use EMM as the tracker of SiamMOT, pre-train using *image training* on CrowdHuman and train on MOT17 using *video training*.

We obtain our results by submitting SiamMOT predictions to the official evaluation server of the challenge[1]. The results show that SiamMOT outperforms all previous methods, including the popular Tracktor++ v2 (+9.4 MOTA) and state-of-the-art CenterTrack [76] (+4.4 MOTA).

Note how SiamMOT models instance's motion with region-based features while CenterTrack uses point-based features. As recent research shows [44, 55, 67], region-based features are consistently better for instance recognition and localization. We conjecture this is also true for instance tracking. In addition, CenterTrack implicitly learns to infer the instance's motion in a similar way to the proposed IMM, which is not as good as EMM, as shown in Sec. 5, and by a large body of research in single-object tracking [22, 34, 35, 54].

TAO-person (Tab. 5). We report results on the validation set similar to [14]. We train a SiamMOT with EMM using *image training* on MSCOCO and CrowdHuman datasets. SiamMOT outperforms state-of-the-art Tracktor++ by a significant 4.4 TrackAP@0.5. As pointed out in [14], linking tracklets with person re-identification embeddings is important in the TAO dataset, as there are a number of videos where people are moving in and out of camera view, which, in this case, is beyond the capability of instance-level motion modelling. Thus, we evaluate SiamMOT+ that merges tracklets with an off-the-shelf person re-id model, the one used in Tracktor++ [5]. Thanks to this,

SiamMOT+ sets new state-of-the-arts on the challenging TAO-person dataset. Although Tracktor++ gains a large 8% TrackAP@0.5 boost by re-id linking, we observe a less significant improvement for SiamMOT. This is because our motion model is already capable of linking challenging cases in TAO, reducing the cases where re-id linking is necessary.

HiEve challenge (Tab. 6). Finally, to further show the strength of SiamMOT, we present results on the recently released Human in Events (HiEve) dataset [41], hosted at the HiEve Challenge at ACM MM'20 [40]. The dataset consists of 19 training and 13 test videos with duration ranging from 30 to 150 seconds, and the videos mainly feature surveillance scenes in subways, restaurants, shopping malls and outdoor streets. We report results on the test set using the publicly released detections. We jointly train a SiamMOT with EMM on CrowdHuman and HiEve training videos. We obtain our results by submitting its predictions to the official evaluation server of the challenge[2]. We submit two sets of results, one obtained with a lightweight DLA-34 backbone and one with a heavier-weight DLA-169. While the former already matches the top performance in ACM MM'20 HiEve Challenge [40], the latter beats all winning methods that are heavily tuned for the challenge.

7. Conclusion

We presented a region-based MOT network – SiamMOT, which detects and associates object instances simultaneously. In SiamMOT, detected instances are temporally linked by a Siamese tracker that models instance motion across frames. We found that the capability of the tracker within SiamMOT is particularly important to the MOT performance. We applied SiamMOT to three different multi-person tracking datasets, and it achieved top results on all of them, demonstrating that SiamMOT is a state-of-the-art tracking network. Although SiamMOT has proven to work well on person tracking, its framework can be easily adapted to accommodate multi-class multi-object tracking, and we plan to explore this direction in the future.

---

[1] https://motchallenge.net/

[2] http://humaninevents.org/

References

[1] Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. In *CVPR 2011*. IEEE, 2011.

[2] Yutong Ban, Sileye Ba, Xavier Alameda-Pineda, and Radu Horaud. Tracking multiple persons based on a variational bayesian model. In *European Conference on Computer Vision*, pages 52–67. Springer, 2016.

[3] Jerome Berclaz, Francois Fleuret, and Pascal Fua. Robust people tracking with global trajectory optimization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006.

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011.

[5] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019.

[6] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.

[7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[8] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

[9] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, 2015.

[11] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *European Conference on Computer Vision*. Springer, 2010.

[12] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

[13] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019.

[14] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European Conference on Computer Vision*, 2020.

[15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[16] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019.

[17] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[18] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019.

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

[20] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[21] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[22] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6269–6277, 2020.

[23] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017.

[24] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019.

[25] David Hall and Pietro Perona. Fine-grained classification of pedestrians in video: Benchmark and state of the art. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5482–5491, 2015.

[26] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, 2018.

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[28] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.

[29] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. *arXiv preprint arXiv:1705.08314*, 2017.

[30] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. In *CVPRW*, 2017.

[31] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[32] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *TPAMI*, 42(1):140–153, 2018.

[33] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *ICCV*, 2015.

[34] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[35] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[36] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[38] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[40] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Guo-Jun Qi, Rui Qian, Tao Wang, Nicu Sebe, Ning Xu, Hongkai Xiong, et al. ACM MM Grand Challenge on Large-scale Human-centric Video Analysis in Complex Events. `http://humaninevents.org/ACM_welcome.html`, 2020.

[41] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Guo-Jun Qi, Rui Qian, Tao Wang, Nicu Sebe, Ning Xu, Hongkai Xiong, et al. Human in events: A large-scale benchmark for human-centric video analysis in complex events. *arXiv preprint arXiv:2005.04490*, 2020.

[42] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[43] Jinlong Peng, Yueyang Gu, Yabiao Wang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dense scene multiple object tracking with box-plane matching. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4615–4619, 2020.

[44] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun. Borderdet: Border feature for dense object detection. In *European Conference on Computer Vision*. Springer, 2020.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[46] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, 2018.

[47] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[48] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[49] Hao Sheng, Yang Zhang, Jiahui Chen, Zhang Xiong, and Jun Zhang. Heterogeneous association graph fusion for target association in multiple object tracking. *TCSVT*, 29(11):3269–3280, 2018.

[50] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016.

[51] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[52] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):104–119, 2019.

[53] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017.

[54] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[55] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.

[56] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019.

[57] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017.

[58] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European conference on computer vision*, pages 135–153. Springer, 2016.

[59] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[60] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[61] Ancong Wu, Chengzhi Lin, Bogao Chen, Weihao Huang, Zeyu Huang, and Wei-Shi Zheng. Transductive multi-object tracking in complex events by interactive self-training. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4620–4624, 2020.

[62] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[63] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In

Proceedings of the IEEE international conference on computer vision, pages 4705–4713, 2015.

[64] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, 2019.

[65] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixe, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. *arXiv preprint arXiv:1906.06618*, 2019.

[66] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[67] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019.

[68] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

[69] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[70] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[71] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012.

[72] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

[73] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *CVPR*, 2019.

[74] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019.

[75] Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. In *ICCV*, 2019.

[76] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020.

[77] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *European Conference on Computer Vision*. Springer, 2020.

[78] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.