本文提出的 Tracktor仅通过一个目标检测器即可完成MOT任务。对于给定的帧t包含两个主要的处理步骤，如图1，用蓝色和红色表示。首先（蓝色部分， 对应轨迹bbox回归更新），将t-1帧已有的轨迹边界框作为第t帧该轨迹的起始边界框，进行回归对齐即可得到当前帧对应轨迹的bbox（这里的理论支持前提是高帧率视频下，前一帧和后一帧位置变化不大）。然后，将新bbox的位置相应对象分类分数用于kill掉可能被遮挡的track。其次（红色部分， 新轨迹的加入），对于新轨迹的出现，检测器仍然提供t帧的一组检测D_t，如果其中某些检测结果没有与已存在的轨迹集合B_t的任何边界框的IOU超过一定阈值，则初始化这些检测为新轨迹。
以上是Tracktor的核心内容，Tracktor++就是加入ReID和运动模型。加入Motion model是针对之前的假设：两帧之间的变化不是很大，在有些情况下并不成立的情况考虑的。在极端的情况下，BBox 从 frame t-1 在第 t 帧中可能根本不包含目标物体了。所以，作者设计了两种 motion model 来改善 BBox 在将来帧中的位置。对于运动相机，作者采用相机运动补偿（camera motion compensation，CMC）的方式进行缓解。作者采用图像配准的方式来对齐视频帧，用的是 增强型相关系数（ECC）最大化。对于低帧率的视频，作者采用匀速假设(CV)。Re-identification是为了让 tracker能够保持跟踪那些被遮挡的物体，作者提出利用 short-term re-ID 的方式（借助 Siamese Network 来进行 appearance feature 的匹配）来改善效果。（需要单独训练。就是把原本没有tracking的bbox拿出来取appearance feature，这一部分只针对那些被中断的track做。）为了达到这个目标，作者将删除了的目标，存储固定帧数的样本。然后将这些样本和新检测的目标在 embedding space 进行重识别。为了最大程度地降低错误reID的风险，我们仅考虑IoU足够大的成对的停用边界框和新边界框。

当然，还有一个要做的事情就是，什么时候tracking停止：（1）如果下一帧的得到的classification score低于某个阈值，也即可能object移除视野或者被其他非object遮挡（2）通过对所有剩余的Bt及其对应的score应用非最大抑制（NMS）来处理对象之间的遮挡，NMS会设置一个阈值。

这个简单的利用Object detector直接做tracking的思路看似不错，但也许只对光照条件比较好的摄像头图像域跟踪有用（i.e.只对满足这类条件的数据集有用），因为一个显然的问题是，没有考虑到怎么去handle false detection的case。而对于光照条件比较好的摄像头图像域跟踪的问题，基本可能不会出现false detection。

Abstract

The problem of tracking multiple objects in a video sequence poses several challenging tasks. For tracking-by-detection, these include object re-identification, motion prediction and dealing with occlusions. We present a tracker (without bells and whistles) that accomplishes tracking without specifically targeting any of these tasks, in particular, we perform no training or optimization on tracking data. To this end, we exploit the bounding box regression of an object detector to predict the position of an object in the next frame, thereby converting a detector into a Tracktor. We demonstrate the potential of Tracktor and provide a new state-of-the-art on three multi-object tracking benchmarks by extending it with a straightforward re-identification and camera motion compensation.

We then perform an analysis on the performance and failure cases of several state-of-the-art tracking methods in comparison to our Tracktor. Surprisingly, none of the dedicated tracking methods are considerably better in dealing with complex tracking scenarios, namely, small and occluded objects or missing detections. However, our approach tackles most of the easy tracking scenarios. Therefore, we motivate our approach as a new tracking paradigm and point out promising future research directions. Overall, Tracktor yields superior tracking performance than any current tracking method and our analysis exposes remaining and unsolved tracking challenges to inspire future research directions.

1. Introduction

Scene understanding from video remains one of the big challenges of computer vision. Humans are often the center of attention in a scene, which leads to the fundamental problem of detecting and tracking them in a video. *Tracking-by-detection* has emerged as the preferred paradigm to solve the problem of tracking multiple objects as it simplifies the task by breaking it into two steps: (i) detecting object locations independently in each frame, (ii) form tracks by linking corresponding detections across time. The linking step, or *data association*, is a challenging task on its own, due to missing and spurious detections, occlusions, and target interactions in crowded environments. To address these issues, research in this area has produced increasingly complex models achieving only marginally better results, e.g., multiple object tracking accuracy has only improved 2.4% in the last two years on the MOT16 [45] benchmark.

In this paper, we push tracking-by-detection to the limit by using only an object detection method to perform tracking. We show that one can achieve state-of-the-art tracking results by training a neural network only on the task of *detection*. As indicated by the blue arrows in Figure 1, the regressor of an object detector such as Faster-RCNN [52] is sufficient to construct object trajectories in a multitude of challenging tracking scenarios. This raises an interesting question that we discuss in this paper: If a *detector* can solve most of the tracking problems, what are the real situations where a dedicated *tracking* algorithm is necessary? We hope our work and the presented *Tracktor* allows researchers to focus on the still unsolved critical challenges of multi-object tracking.

This paper presents four main contributions:

- We introduce the Tracktor which tackles multi-object tracking by exploiting the regression head of a detector to perform temporal realignment of object bounding boxes.
- We present two simple extensions to Tracktor, a re-identification Siamese network and a motion model. The resulting tracker yields state-of-the-art performance in three challenging multi-object tracking benchmarks.
- We conduct a detailed analysis on failure cases and challenging tracking scenarios, and show none of the dedicated tracking methods perform substantially better than our regression approach.
- We propose our method as a new tracking paradigm which exploits the detector and allows researchers to focus on the remaining complex tracking challenges. This includes an extensive study on promising future research directions.

*Contributed equally. Correspondence to: tim.meinhardt@tum.de

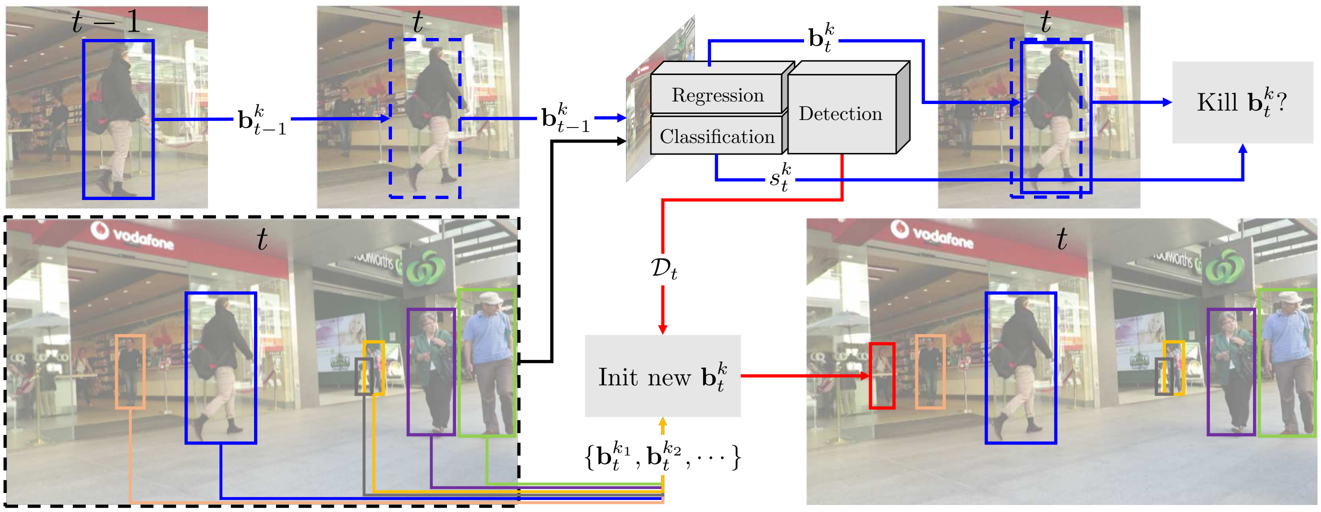


Figure 1: The presented Tracktor accomplishes multi-object tracking only with an object detector and consists of two primary processing steps, indicated in blue and red, for a given frame t. First, the regression of the object detector aligns already existing track bounding boxes $\mathbf{b}_{t-1}^k$ of frame $t-1$ to the object's new position at frame t. The corresponding object classification scores s_t^k of the new bounding box positions are then used to kill potentially occluded tracks. Second, the object detector (or a given set of public detections) provides a set of detections $\mathcal{D}_t$ of frame t. Finally, a new track is initialized if a detection has no substantial Intersection over Union with any bounding box of the set of active tracks $B_t = \{\mathbf{b}_t^{k_1}, \mathbf{b}_t^{k_2}, \cdots\}$.

1.1. Related work

Several computer vision tasks such as surveillance, activity recognition or autonomous driving rely on object trajectories as input. Despite the vast literature on multi-object tracking [42, 38], it still remains a challenging problem, especially in crowded environments where occlusions and false detections are common. Most state-of-the-art works follow the tracking-by-detection paradigm which heavily relies on the performance of the underlying detection method.

Recently, neural network based detectors have clearly outperformed all other methods for detection [33, 52, 50]. The family of detectors that evolved to Faster-RCNN [52], and further detectors such as SDP [63], rely on object proposals which are passed to an object classification and a bounding box regression head of a neural network. The latter refines bounding boxes to fit tightly around the object. In this paper, we show that one can rethink the use of this regressor for tracking purposes.

Tracking as a graph problem. The data association problem deals with keeping the identity of the tracked objects given the available detections. This can be done on a frame by frame basis for online applications [5, 15, 48] or track-by-track [3]. Since video analysis can be done offline, batch methods are preferred since they are more robust to occlusions. A common formalism is to represent the problem as a graph, where each detection is a node, and edges indicate a possible link. The data association can then be formulated as maximum flow [4] or, equivalently, minimum cost problem with either fixed costs based on distance [26, 49, 66], including motion models [39], or learned costs [36]. Alternative formulations typically lead to more involved optimization problems, including minimum cliques [65], general-purpose solvers like MCMC [64] or multi-cuts [59]. A recent trend is to design ever more complex models which include other vision input such as reconstruction for multi-camera sequences [40, 60], activity recognition [12], segmentation [46], keypoint trajectories [10] or joint detection [59]. In general, the significantly higher computational costs do not translate to significantly higher accuracy. In fact, in this work, we show that we can outperform all graph-based trackers significantly while keeping the tracker online. Even within a graphical model optimization, one needs to define a measure to identify whether two bounding boxes belong to the same person or not. This can be done by analyzing either the appearance of the pedestrian, or its motion.

Appearance models and re-identification. Discriminating and re-identifying (reID) objects by appearance is in particular a problem in crowded scenes with many object-object occlusions. In the exhaustive literature that uses appearance models or reID methods to improve multi-object tracking, color-based models are very common [31]. However, these are not always reliable for pedestrian tracking, since people can wear very similar clothes, and color statistics are often contaminated by background pixels and illumination changes. The authors of [34] borrow ideas from person re-identification and adapt them to "re-identify" targets during tracking. In [62], a CRF model is learned to better distinguish pedestrians with similar appearance. Both appearance and short-term motion in the form of optical flow can be used as input to a Siamese neural network to decide whether two boxes belong to the same track or not [35]. Recently, [54] showed the importance of learned reID features for multi-object tracking. We confirm this view in our experiments.

Motion models and trajectory prediction. Several works resort to motion to discriminate between pedestrians, especially in highly crowded scenes. The most common assumption is the one of constant velocity (CVA) [11, 2], but pedestrian motion gets more complex in crowded scenarios for which researchers have turned to the more expressive Social Force Model [57, 48, 61, 39]. Such a model can also be learned from data [36]. Deep Learning has been extensively used to learn social etiquette in crowded scenarios for trajectory prediction [39, 1, 55]. [67] use single object tracking trained networks to create tracklets for further post-processing into trajectories. Recently, [7, 51] proposed to use reinforcement learning to predict the position of an object in the next frame. While [7] focuses on single object tracking, the authors of [51] train a multi-object pedestrian tracker composed of a bounding box predictor and a decision network for collaborative decision making between tracked objects.

Video object detection. Multi-object tracking without frame-to-frame identity prediction is a subproblem usually referred to as video object detection. In order to improve detections, many methods exploit spatio-temporal consistencies of object positions. Both [28] and [27] generate multi-frame bounding box tuplet proposals and extract detection scores and features with a CNN and LSTM, respectively. Recently, the authors of [47] improve object detections by applying optical flow to propagate scores between frames. Eventually, [18] proposes to solve the tracking and detection problem jointly. They propose a network which processes two consecutive frames and exploits tracking ground truth data to improve detection regression, thereby, generating two-frame tracklets. With a subsequent offline method, these tracklets are combined to multi-frame tracks. However, we show that our regression tracker is not only online, but superior in dealing with object occlusions. In particular, we do not only temporally align detections, but preserve their identity.

2. A detector is all you need

We propose to convert a *detector* into a *Tracktor* performing multiple object tracking. Several CNN-based detection algorithms [52, 63] contain some form of bounding box refinement through regression. We propose an exploitation of such a regressor for the task of tracking. This has two key advantages: (i) we do not require any tracking specific training, and (ii) we do not perform any complex optimization at test time, hence our tracker is online. Furthermore, we show that our method achieves state-of-the-art performance on several challenging tracking scenarios.

2.1. Object detector

The core element of our tracking pipeline is a regression-based detector. In our case, we train a Faster R-CNN [52] with ResNet-101 [22] and Feature Pyramid Networks (FPN) [41] on the MOT17Det [45] pedestrian detection dataset.

To perform object detection, Faster R-CNN applies a Region Proposal Network to generate a multitude of bounding box proposals for each potential object. Feature maps for each proposal are extracted via Region of Interest (RoI) pooling [21], and passed to the classification and regression heads. The classification head assigns an object score to the proposal, in our case, it evaluates the likelihood of the proposal showing a pedestrian. The regression head refines the bounding box location tightly around an object. The detector yields the final set of object detections by applying non-maximum-suppression (NMS) to the refined bounding box proposals. Our presented method exploits the aforementioned ability to regress and classify bounding boxes to perform multi-object tracking.

2.2. Tracktor

The challenge of multi-object tracking is to extract the spatial and temporal positions, i.e., trajectories, of k objects given a frame by frame video sequence. Such a trajectory is defined as a list of ordered object bounding boxes $T_k = \{\mathbf{b}^k_{t_1}, \mathbf{b}^k_{t_2}, \cdots\}$, where a bounding box is defined by its coordinates $\mathbf{b}^k_t = (x, y, w, h)$, and t represents a frame of the video. We denote the set object bounding boxes in frame t with $B_t = \{\mathbf{b}^{k_1}_t, \mathbf{b}^{k_2}_t, \cdots\}$. Note, that each T_k or B_t can contain less elements than the total number of frames or trajectories in a sequence, respectively. At $t = 0$, our tracker initializes tracks from the first set of detections $\mathcal{D}_0 = \{\mathbf{d}^1_0, \mathbf{d}^2_0, \cdots\} = B_0$. In Figure 1, we illustrate the two subsequent processing steps (the nuts and bolts of our method) for a given frame t for all $t > 0$, namely, the bounding box regression and track initialization.

Bounding box regression. The first step, denoted with blue arrows, exploits the bounding box regression to extend active trajectories to the current frame t. This is achieved by regressing the bounding box $\mathbf{b}^k_{t-1}$ of frame $t-1$ to the object's new position $\mathbf{b}^k_t$ at frame t. In the case of Faster R-CNN, this corresponds to applying RoI pooling on the features of the current frame but with the previous bounding box coordinates. Our assumption is that the target has moved only slightly between frames, which is usually ensured from high frame rates (see Section B.5 of the supplementary for a frame rate robustness evaluation of Tracktor).The identity is automatically transferred from the previous to the regressed bounding box, effectively creating a trajectory. This is repeated for all subsequent frames.

After the bounding box regression, our tracker considers two cases for killing (deactivating) a trajectory: (i) an object leaving the frame or occluded by a non-object is killed if its new classification score s_t^k is below σ_{active} and (ii) occlusions between objects are handled by applying non-maximum suppression (NMS) to all remaining B_t and their corresponding scores with an Intersection over Union (IoU) threshold λ_{active}.

Bounding box initialization. In order to account for new targets, the object detector also provides the detections $\mathcal{D}_t$ for the entire frame t. This second step, indicated in Figure 1 with red arrows, is analogous to the first initialization at $t = 0$. But a detection from $\mathcal{D}_t$ starts a trajectory only if the IoU with any of the already active trajectories $\mathbf{b}_t^k$ is smaller than λ_{new}. That is, we consider a detection for a new trajectory only if it is covering a potentially new object that is not explained by any trajectory. It should be noted again that our Tracktor does not require any tracking specific training or optimization and solely relies on an object detection method. This allows us to directly benefit from improved object detection methods and, most importantly, enables a comparatively cheap transfer to different tracking datasets or scenarios in which no ground truth tracking but only detection data is available.

2.3. Tracking extensions

In this section, we present two straightforward extensions to our vanilla Tracktor: a motion model and a re-identification algorithm. Both are aimed at improving identity preservation across frames and are common examples of techniques used to enhance, e.g., graph-based tracking methods [39, 62, 35].

Motion model. Our previous assumption that the position of an object changes only slightly from frame to frame does not hold in two scenarios: large camera motion and low video frame rates. In extreme cases, the bounding boxes from frame $t-1$ might not contain the tracked object in frame t at all. Therefore, we apply two types of motion models that will improve the bounding box position in future frames. For sequences with a moving camera, we apply a straightforward camera motion compensation (CMC) by aligning frames via image registration using the Enhanced Correlation Coefficient (ECC) maximization as introduced in [16]. For sequences with comparatively low frame rates, we apply a constant velocity assumption (CVA) for all objects as in [11, 2].

Re-identification. In order to keep our tracker online, we suggest a short-term re-identification (reID) based on appearance vectors generated by a Siamese neural network [6, 25, 54]. To that end, we store killed (deactivated) tracks in their non-regressed version $\mathbf{b}_{t-1}^k$ for a fixed number of F_{reID} frames. We then compare the distance in the embedding space of the deactivated with the newly detected tracks and re-identify via a threshold. The embedding space distance is computed by a Siamese CNN and appearance feature vectors for each of the bounding boxes. It should be noted that the reID network is indeed trained on tracking ground truth data. To minimize the risk of false reIDs, we only consider pairs of deactivated and new bounding boxes with a sufficiently large IoU. The motion model is continuously applied to the deactivated tracks.

3. Experiments

We demonstrate the tracking performance of our proposed Tracktor tracker as well as its extension *Tracktor++* on several datasets focusing on pedestrian tracking. [1] In addition, we perform an ablation study of the aforementioned extensions and further show that our tracker outperforms state-of-the-art methods in tracking accuracy and excels at identity preservation.

MOTChallenge. The multi-object tracking benchmark MOTChallenge [2] consists of several challenging pedestrian tracking sequences, with frequent occlusions and crowded scenes. Sequences vary in their angle of view, size of objects, camera motion and frame rate. The challenge contains three separate tracking benchmarks, namely *2D MOT 2015* [37], *MOT16* and *MOT17* [45]. The MOT17 test set includes a total of 7 sequences each of which is provided with three sets of public detections. The detections originate from different object detectors each with increasing performance, namely DPM [19], Faster R-CNN [52] and SDP [63]. Our object detector is trained on the MOT17Det [45] detection benchmark which contains the same images as MOT17. The MOT16 benchmark also contains the same sequences as MOT17 but only provides DPM public detections. The 2D MOT 2015 benchmark provides ACF [14] detections for 11 sequences. The complexity of the tracking problem requires several metrics to measure different aspects of a tracker's performance. The Multiple Object Tracking Accuracy (MOTA) [29] and ID F1 Score (IDF1) [53] quantify two of the main aspects, namely, object coverage and identity.

Public detections. For a fair comparison with other tracking methods, we perform all experiments with the public detections provided by MOTChallenge. That is, all methods compared in this paper, including our approach and its extension, process the same precomputed frame by frame detections. For our method, a new trajectory is *only* initialized from a public detection bounding box, i.e., we *never*

[1] Tracktor code: https://git.io/fjQr8.

[2] The MOTChallenge web page: https://motchallenge.net.

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
D&T [18]	50.1	24.9	23.1	27.1	3561	52481	2715
Tracktor-no-FPN	57.4	58.7	30.2	22.5	2821	45042	1981
Tracktor	61.5	61.1	33.5	**20.7**	367	42903	1747
Tracktor+reID	61.5	62.8	33.5	**20.7**	367	42903	921
Tracktor+CMC	**61.9**	64.1	**35.3**	21.4	**323**	**42454**	458
Tracktor++ (reID + CMC)	**61.9**	**64.7**	**35.3**	21.4	**323**	**42454**	**326**

Table 1: This ablation study illustrates multiple aspects on the performance of our Tracktor. In particular the improvements from extending it with tracking specific methods, i.e., a short-term bounding box re-identification and camera motion compensation by frame alignment. The combination yields the Tracktor++ tracker. We evaluated only on the Faster R-CNN set of MOT17 public detections. The arrows indicate low or high optimal metric values.

use our object detector to detect a new bounding box. We only apply the bounding box regressor and classifier to obtain new $\mathbf{b}_t^k$ and s_t^k, respectively. The MOTChallenge public benchmark includes multiple methods [30, 9, 13] which classify the given detections with trained neural networks, hence, we consider our processing of the given detections also as *public*.

3.1. Ablation study

The ablation study on the MOT17 [45] training set in Table 1 is intended to show three aspects: (i) the superiority of our approach when applying a detector for tracking, (ii) the potential from an improved object detection method and (iii) improvements from extending our vanilla Tracktor with tracking specific methods, namely, re-identification (reID) and camera motion compensation (CMC). It should be noted, that although MOT17Det and MOT17 contain the same images, we refrained from a cross-validation on the training set as our vanilla Tracktor was never trained on tracking ground truth data. The video object detector and tracker *D&T* [18] trains a detector on tracking ground truth data which generates two-frame tracklets. However, despite a subsequent offline dynamic programming track generation their detector-based tracker is inferior to our online regression-based track generation over multiple frames. In addition, we demonstrate the potential of our framework with respect to improved detection methods by showing the tracking performance of *Tracktor-no-FPN*, i.e., our approach and a Faster R-CNN without Feature Pyramid Networks (FPN) [41]. Despite the simple nature of our extensions to Tracktor++, their contribution is significant towards the drastic reduction of identity switches and an increment of the IDF1 measure. In the next section, we show that this effect successfully translates to a comparison with other state-of-the-art methods on the test set.

	Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
MOT17	Tracktor++	**53.5**	52.3	19.5	36.6	**12201**	248047	2072
	eHAF [58]	51.8	**54.7**	**23.4**	37.9	33212	**236772**	1834
	FWT [23]	51.3	47.6	21.4	35.2	24101	247921	2648
	jCC [30]	51.2	54.5	20.9	37.0	25937	247822	**1802**
	MOTDT17 [9]	50.9	52.7	17.5	35.7	24069	250768	2474
	MHT_DAM [32]	50.7	47.2	20.8	36.9	22875	252889	2314
MOT16	Tracktor++	**54.4**	**52.5**	19.0	**36.9**	**3280**	**79149**	682
	HCC [44]	49.3	50.7	17.8	39.9	5333	86795	**391**
	LMP [59]	48.8	51.3	18.2	40.1	6654	86245	481
	GCRA [43]	48.2	48.6	12.9	41.1	5104	88586	821
	FWT [23]	47.8	44.3	**19.1**	38.2	8886	85487	852
	MOTDT [9]	47.6	50.9	15.2	38.3	9253	85431	792
2D MOT 2015	Tracktor++	**44.1**	46.7	18.0	**26.2**	6477	**26577**	1318
	AP_HWDPL_p [8]	38.5	**47.1**	8.7	37.4	**4005**	33203	586
	AMIR15 [56]	37.6	46.0	15.8	26.8	7933	29397	1026
	JointMC [30]	35.6	45.1	**23.2**	39.3	10580	28508	457
	RAR15pub [17]	35.1	45.4	13.0	42.3	6771	32717	**381**

Table 2: We compare our online multi-object tracker Tracktor++ with other modern tracking methods. As a result, we achieve a new state-of-the-art in terms of MOTA for public detections on all three MOTChallenge benchmarks. The arrows indicate low or high optimal metric values.

3.2. Benchmark evaluation

We evaluate the performance of our Tracktor++ on the test set of the respective benchmark, without any training or optimization on the tracking train set. Table 2 presents the overall results accumulated over all sequences, and for MOT17 over all three sets of public detections. For our comparison, we only consider officially published and peer-reviewed entries in the MOTChallenge benchmark. Our supplementary material provides a detailed summary of all results on individual sequences. For all sequences, camera motion compensation (CMC) and reID are used. The only low frame rate sequence is the 2D MOT 2015 *AVG-TownCentre*, for which we apply the aforementioned constant velocity assumption (CVA). For the two autonomous driving sequences, originally from the KITTI [20] benchmark, we apply the rotation as well as translation camera motion compensation. Note, we use the same Tracktor++ tracker, trained on MOT17Det object detections, for all benchmarks. As we show, it is able to achieve a new state-of-the-art in terms of MOTA on all three challenges.

In particular, our results on MOT16 demonstrate the ability of our tracker to cope with detections of comparatively minor performance. Due to the nature of our tracker and the robustness of the frame by frame bounding box regression, we outperform all other trackers on MOT16 by a large margin, specifically in terms of false negatives (FN) and identity preserving (IDF1). It should be noted, that we also provide a new state-of-the-art on 2D MOT 2015, even though the characteristics of the scenes are very different from MOT17. We do not use MOT15 training sequences, which further illustrates the generalization strength of our tracker.

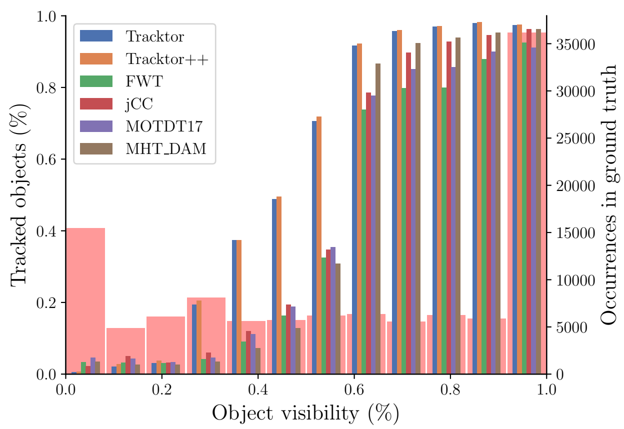


Figure 2: We illustrate the ratio of tracked objects with respect to their visibility evaluated on the Faster R-CNN public detections. The results clearly demonstrate that none of the presented more sophisticated methods achieves superior performance to our approach. This is especially noticeable for highly occluded boxes. The transparent red bars indicate the ground truth distribution of visibilities.

Method	Online	Graph	reID	Appearance model	Motion model	Other
Tracktor	×					
Tracktor++	×		×		Camera	
FWT [23]		Dense				Face detection
jCC [30]		Dense				Point trajectories
MOTDT17 [9]	×		×	×	Kalman	
MHT_DAM [32]		Sparse		×	Kalman	

Table 3: A summary of the fundamental characteristics of our methods and other state-of-the-art trackers.

4. Analysis

The superior performance of our tracker without any tracking specific training or optimization demands a more thorough analysis. Without sophisticated tracking methods, it is not expected to excel in crowded and occluded, but rather only in benevolent, tracking scenarios. Which begs the question whether more common tracking methods fail to specifically address these complex scenarios as well. Our experiments and the subsequent analysis ought to demonstrate the strengths of our approach for easy tracking scenarios and motivate future research to focus on remaining complex tracking problems. In particular, we question the common execution of tracking-by-detection and suggest a new tracking paradigm. The subsequent analysis is conducted on the MOT17 training data and we compare all top performing methods with publicly shared data.

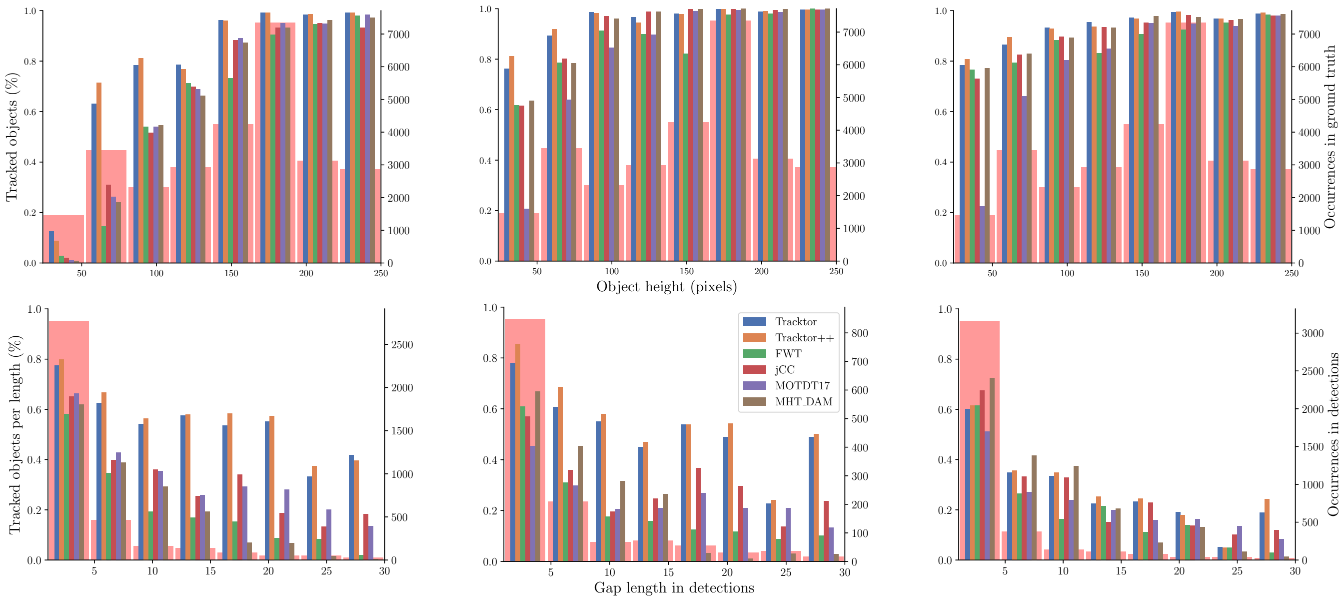
4.1. Tracking challenges

For a better understanding of our tracker, we want to analyse challenging tracking scenarios and compare its strengths and weaknesses to other trackers. To this end, we summarize their fundamental characteristics in Table 3. FWT [23] and jCC [30] both apply a dense offline graph optimization on all detections in a given sequence. In contrast, MHT_DAM [32] limits its optimization to a sparse forward view of hypothetical trajectories.

Object visibility. Intuitively, we expect diminished tracking performance for object-object or object-non-object occlusions, i.e., for targets with diminished visibility. In Figure 2, we compare the ratio of successfully tracked bounding boxes with respect to their visibility. The transparent red bar indicates the occurrences of ground truth bounding boxes for each visibility, and illustrates the proportionate impact on the overall performance of the trackers. Our method achieves superior performance even for partially occluded bounding boxes with visibilities as low as 0.3. Neither the identify preserving aspects of MHT_DAM and MOTDT17 [9] nor the offline interpolation capabilities of MHT_DAM and jCC seem to successfully tackle highly occluded objects. The high MOTA values in Table 2 are largely due to the unbalanced distribution of ground truth visibilities. As expected, our extended version only achieves minor improvements over our vanilla Tracktor.

Object size. In view of the large fraction of visible but not tracked objects in Figure 2, we argue that the trackability of an object is not only dependent on its visibility, but also its size. Therefore, we conduct the same comparison as for the visibility but for the size of an object. In the first row of Figure 3, we assume the height of a pedestrian to be proportional to its size and compare on all three MOT17 public detection sets. All methods performed similarly well for object heights larger than 250 pixels. To demonstrate their shortcomings even for highly visible objects, we only compare objects with a visibility larger than 0.9. As expected, the trackability of an object decreases drastically with its size across all three detection sets. Our tracker shows its strength in compensating for insufficient DPM and Faster R-CNN detections for all object sizes. All methods except MOTDT17 benefit from the additional small detections provided by SDP. For our tracker this is largely due to the Feature Pyramid Network extension of our Faster-RCNN detector. However, the learned appearance model and reID of the online MOTDT17 method seem generally vulnerable to small detections. Appearance models generally suffer from small object sizes and few observed pixels. In conclusion, except from our compensation of inferior detections none of the trackers exhibit a notably better performance with respect to varying object sizes.

Robustness to detections. The performance of tracking-by-detection methods with respect to visibility and size is inherently limited by the robustness of the underlying detection method. However, as observed for the object size, trackers differ in their ability to cope with, or benefit from, varying quality of detections. In the second row of Figure 3, we quantify this ability in terms of detection gaps



(a) DPM detections

(b) Faster R-CNN detections

(c) SDP detections

Figure 3: The two rows illustrate the ratio of tracked objects with respect to: (i) object heights and (ii) the length of gaps in the provided public detections. The transparent red bars indicate the ground truth distribution of heights and gap lengths in the detections, respectively. To demonstrate the shortcomings of the presented trackers we limited the height comparison to objects with visibility greater or equal than 0.9. Tracks that are not detected at all are not considered as a gap. Hence, SPD generates the most gaps. For it also provides the most detections.

on their coverage by the tracker. We define a detection gap as part of a ground truth trajectory that was at least once detected, and compare coverage of each gap vs. the gap length. Intuitively, long gaps are harder to compensate for, as the online or offline tracker has to perform a longer hallucination or interpolation, respectively. We indicated the occurrences of gap lengths over the respective set of detections in transparent red. For DPM and Faster R-CNN detections, two solutions lead to notable gap coverage: (i) offline interpolation such as in jCC, or (ii) motion prediction with Kalman filter and reID as in MOTDT. Compared to the graph-based jCC method, the online MOTDT17 method excels at covering particularly long gaps. However, none of these dedicated tracking methods yields similar robustness to our frame by frame regression tracker, which achieves far superior coverage. This holds especially true for long detection gaps with more than 15 frames. Offline methods benefit the most from improved SDP detections and neither our nor the MOTDT17 tracker convince with a notable gap length robustness.

Identity preservation. The results of our Tracktor++ summarized in Table 2 indicate an identity preservation performance in terms of IDF1 and identity switches comparable with dedicated tracking methods. This is achieved without any offline graph optimization as in jCC [30] or eHAF [58]. In particular, MOTDT17, which applies a sophisticated appearance model and reID, is not substantially superior to our regression tracker and its comparatively simple extensions. However, our method excels in reducing the number of false positives in MOT17 as well as MOT16. In addition, we have shown that our Tracktor is capable of incorporating additional identity preserving extension.

4.2. Oracle trackers

We have shown that none of the dedicated tracking methods specifically targets challenging tracking scenarios, i.e., objects under heavy occlusions or small objects. We therefore want to motivate our Tracktor as a new tracking paradigm. To this end, we analyse our performance twofold: (i) the impact of the object detector on the killing policy and bounding box regression, (ii) identify performance upper bounds for potential extensions to our Tracktor. In Table 4, we present several oracle trackers by replacing parts of our algorithm with ground truth information. If not mentioned otherwise, all other tracking aspects are handled by our vanilla Tracktor. Their analysis should provide researchers with useful insights regarding the most promising research directions and extensions of our Tracktor.

Method	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	ID Sw. ↓
Tracktor	61.5	61.1	367	42903	1747
Tracktor++	+0.4	+3.6	-44	-449	-1421
Oracle-Kill	+0.7	-0.7	-178	-694	+129
Oracle-REG	+1.4	+5.6	-218	-1401	-1463
Oracle-MM	+0.9	+5.2	-168	-898	-1332
Oracle-reID	0.0	+10.0	0	0	-1094
Oracle-MM-reID	+0.9	+13.9	-168	-898	-1706
Oracle-MM-reID-INTER	+2.6	+15.9	+3774	-6769	-1680
Oracle-ALL	+10.7	+22.5	-360	-11745	-1743

Table 4: To show the potential of Tracktor and indicate promising future research directions, we present multiple oracle trackers. Each oracle exploits ground truth data for a specific task, simulating, e.g., a perfect re-identification (reID) or motion model (MM). We evaluate only on the Faster R-CNN set of MOT17 public detections and highlight performance gains and losses with respect to the vanilla Tracktor in green and red, respectively. The arrows indicate low or high optimal metric values.

Detector oracles. To simulate a potentially perfect object detector, we introduce two oracles:

- Oracle-Kill: Instead of killing with NMS or classification score we use ground truth information.
- Oracle-REG: Instead of regression, we place the bounding boxes at their ground truth position.

Both oracles yield substantial improvements with respect to MOTA and FP. However, killing by ground truth instead of score deteriorates identity preservation as the regression struggles with otherwise unseen bounding boxes.

Extension oracles. It should be noted, that Tracktor++ with non-perfect extensions already compensates for some of the detector's insufficiencies. The reID and motion model (MM) oracles simulate potential additional performance gains. In order to remain online, these exclude any form of hindsight tracking-gap interpolation.

- Oracle-MM: A motion model places each bounding box at the center of the ground truth in the next frame.
- Oracle-reID: Re-identification is performed with ground truth identities.

As expected, both oracles improve IDF1 and identity switches substantially. The combined Oracle-MM-reID represents the extension upper bound of Tracktor++.

Omniscient oracle. Oracle-ALL performs ground truth killing, regression and reID. We consider its top MOTA of 72.2%, in combination with a high IDF1 and virtually no false positives, as the absolute upper bound of Tracktor with a Faster R-CNN and FPN object detector.

The substantial performance gains from Oracle-MM indicate the potential of extending Tracktor with a sophisticated motion model. In particular, Oracle-MM-reID-INTER suggests a predictive motion model which hallucinates the position of an object through long occlusions. Such a motion model avoids offline post processing and additional false positives from wrong linear occlusion paths caused by long detection gaps and camera movement

4.3. Towards a new tracking paradigm

To conclude our analysis we propose two approaches on how to utilize Tracktor as a starting point for future research directions:

Tracktor with extensions. Apply Tracktor to a given set of detections and extend it with tracking specific methods. Scenarios with large and highly visible objects will be covered by the frame to frame bounding box regression. For the remaining, it seems most promising to implement a hallucinating motion model, taking into account the individual movements of objects. In addition, such a motion predictor reduces the necessity for an advanced killing policy.

Tracklet generation. Analogous to tracking-by-detection, we propose a tracking-by-tracklet approach. Indeed, many algorithms already use tracklets as input [24, 65], as they are richer in information for computing motion or appearance models. However, usually a specific tracking method is used to create these tracklets.We advocate the exploitation of the detector itself, not only to create sparse detections, but frame to frame tracklets. The remaining complex tracking cases ought to be tackled by a subsequent tracking method.

In this work, we have formally defined those hard cases, analyzing the situations in which not only our method but other dedicated tracking solutions fail. And by doing so, we question the current focus of research in multi-object tracking, in particular, the missing confrontation with challenging tracking scenarios.

5. Conclusions

We have shown that the bounding box regressor of a trained Faster-RCNN detector is enough to solve most tracking scenarios present in current benchmarks. A detector converted to Tracktor needs no specific training on tracking ground truth data and is able to work in an online fashion. In addition, we have shown that our Tracktor is extendable with re-identification and camera motion compensation, providing a substantial new state-of-the-art on the MOTChallenge. We analyzed the performance of multiple dedicated tracking methods on challenging tracking scenarios and none yielded substantially better performance compared to our regression based Tracktor. We hope this work establishes a new tracking paradigm, utilizing the object detector's full capabilities.

Acknowledgements. This research was funded by the Humboldt Foundation through the Sofja Kovalevskaja Award.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[2] Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1272, 2011. 3, 4

[3] Jérôme Berclaz, Franois Fleuret, and Pascal Fua. Robust people tracking with global trajectory optimization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 744–750, 2006. 2

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011. 2

[5] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. *IEEE International Conference on Computer Vision (ICCV)*, pages 1515–1522, 2009. 2

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Saeckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *NIPS*, 1993. 4

[7] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time 'actor-critic' tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[8] Long Chen, Haizhou Ai, Chong Shang, Zijie Zhuang, and Bo Bai. Online multi-object tracking with convolutional neural networks. pages 645–649, Sept 2017. 5

[9] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification, 07 2018. 5, 6

[10] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. *ICCV*, 2015. 2

[11] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. *European Conference on Computer Vision (ECCV)*, pages 553–567, 2010. 3, 4

[12] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. *European Conference on Computer Vision (ECCV)*, pages 215–230, 2012. 2

[13] Young chul Yoon, Abhijeet Boragule, Young min Song, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *AVSS*, 2018. 5

[14] Piotr Dollar, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *PAMI*, 36(8):1532–1545, Aug. 2014. 4

[15] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 2

[16] Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *PAMI*, 30(10):1858–1865, 2008. 4

[17] Kuan Fang, Yu Xiang, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. *WACV*, abs/1711.02741, 2017. 5

[18] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3057–3065, 2017. 3, 5

[19] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *pami*, 32:1627–1645, 2009. 4

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[21] Ross B. Girshick. Fast r-cnn. *ICCV*, pages 1440–1448, 2015. 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, abs/1512.03385, 2015. 3

[23] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. *CVPR*, abs/1705.08314, 2017. 5, 6

[24] Roberto Henschel, Laura Leal-Taixé, and Bodo Rosenhahn. Efficient multiple people tracking using minimum cost arborescences. *German Conference on Pattern Recognition (GCPR)*, 2014. 8

[25] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 4

[26] Hao Jiang, Sidney S. Fels, and James J. Little. A linear programming approach for multiple object tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2

[27] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–897, 2017. 3

[28] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 817–825, 2016. 3

[29] Rangachar Kasturi, Dmitry B. Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John S. Garofolo, Rachel Bowers, Matthew Boonstra, Valentina N. Korzhova, and Jing Zhang. Framework for performance evaluation for face, text and vehicle detection and tracking in video: data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(2):319–336, 2009. 4

[30] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple ob-

ject tracking by correlation co-clustering. *PAMI*, pages 1–1, 2018. 5, 6, 7

[31] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James Rehg. Multiple hypothesis tracking revisited: Blending in modern appearance model. *ICCV*, 2015. 2

[32] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4696–4704, Dec 2015. 5, 6

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems (NIPS)*, 2012. 2

[34] Cheng-Hao Kuo and Ram Nevatia. How does person identity recognition help multi-person tracking? *CVPR*, 2011. 2

[35] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: siamese cnn for robust target association. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR). DeepVision: Deep Learning for Computer Vision.*, 2016. 3, 4

[36] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 3

[37] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942*, 2015. 4

[38] Laura Leal-Taixé, Anton Milan, Konrad Schindler, Daniel Cremers, Ian D. Reid, and Stefan Roth. Tracking the trackers: An analysis of the state of the art in multiple object tracking. *CoRR*, abs/1704.02781, 2017. 2

[39] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. *IEEE International Conference on Computer Vision (ICCV) Workshops. 1st Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, 2011. 2, 3, 4

[40] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[41] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3, 5

[42] Wenhan Luo, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A review. *arXiv:1409.7618 [cs]*, Sept. 2014. 2

[43] Cong Ma, Changshui Yang, Fan Yang, Yueqing Zhuang, Ziwei Zhang, Huizhu Jia, and Xiaodong Xie. Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. *ICME*, abs/1804.04555, 2018. 5

[44] Liqian Ma, Siyu Tang, Michael J. Black, and Luc Van Gool. Customized multi-person tracker. 2019. 5

[45] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 1, 3, 4, 5

[46] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian D. Reid. Joint tracking and segmentation of multiple targets. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[47] Melissa Ogden, Hongsheng Li, Jingchen Yan, Xingyu Zeng, Bin Yang, Tannan Xiao, Zhe Wang, Ruohui Wang, Xiaogang Wang, and Wanli Ouyang. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28:2896–2907, 2018. 3

[48] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: modeling social behavior for multi-target tracking. *IEEE International Conference on Computer Vision (ICCV)*, pages 261–268, 2009. 2, 3

[49] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208, 2011. 2

[50] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *CVPR*, 2017. 2

[51] Liangliang Ren, Jiwen Lu, Zifeng Wang, Qi Tian, and Jie Zhou. Collaborative deep reinforcement learning for multi-object tracking. *ECCV*, 2018. 3

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NIPS)*, 2015. 1, 2, 3, 4

[53] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. *ECCV Workshops*, 2016. 4

[54] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. *CVPR*, 2018. 3, 4

[55] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory prediction. *European Conference on Computer Vision (ECCV)*, 2016. 3

[56] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *ICCV*, abs/1701.01909, 2017. 5

[57] Paul Scovanner and Marshall F. Tappen. Learning pedestrian dynamics from the real world. *IEEE International Conference on Computer Vision (ICCV)*, pages 381–388, 2009. 3

[58] Hao Sheng, Yang Zhang, Jiahui Chen, Zhang Xiong, and Jun Zhang. Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 5, 7

[59] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3701–3710, July 2017. 2, 5

[60] Zheng Wu, Thomas H. Kunz, and Margrit Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1185–1192, 2011. 2

[61] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and where are you going? *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352, 2011. 3

[62] Bo Yang and Ram Nevatia. An online learned crf model for multi-target tracking. *CVPR*, 2012. 3, 4

[63] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *CVPR*, pages 2129–2137, 2016. 2, 3, 4

[64] Qian Yu, Gerard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2

[65] Amir R. Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. *ECCV*, 2012. 2, 8

[66] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 2

[67] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming H. Yang. Online multi-object tracking with dual matching attention networks. *ECCV*, 2018. 3