

Tracking without bells and whistles

Philipp Bergmann*

Tim Meinhardt*

Laura Leal-Taixe

Technical University of Munich

Abstract

The problem of tracking multiple objects in a video sequence poses several challenging tasks. For tracking-by-detection these include, among others, 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 an object's new position in the next frame, thereby converting a detector into a Tracktor. We demonstrate the extensibility of our 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 and 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, we show that a cleverly exploited detector can perform better tracking than any current tracking method and expose the real tracking challenges which are still unsolved.

1. Introduction

Scene understanding from video remains one of the big challenges of computer vision. Humans are often the center of attention of 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) linking corresponding detections across time to form trajectories. 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 that achieve only marginally better results, e.g., multiple object tracking accuracy has only improved 2.4% in the last two years on the MOT16 MOTChallenge [44] 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 green arrows in Figure 1, the regressor of an object detector such as Faster-RCNN [49] 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 the tracking problem by exploiting the regression head of a detector to perform temporal realignment of object detections.
- We present two simple extensions to our model, a re-identification Siamese network and a motion model. This tracker largely outperforms all trackers 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 approach.
- We propose two ways to use our method as a new tracking paradigm, to allow researchers to focus on the real tracking challenges, while our Tracktor handles the rest.

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

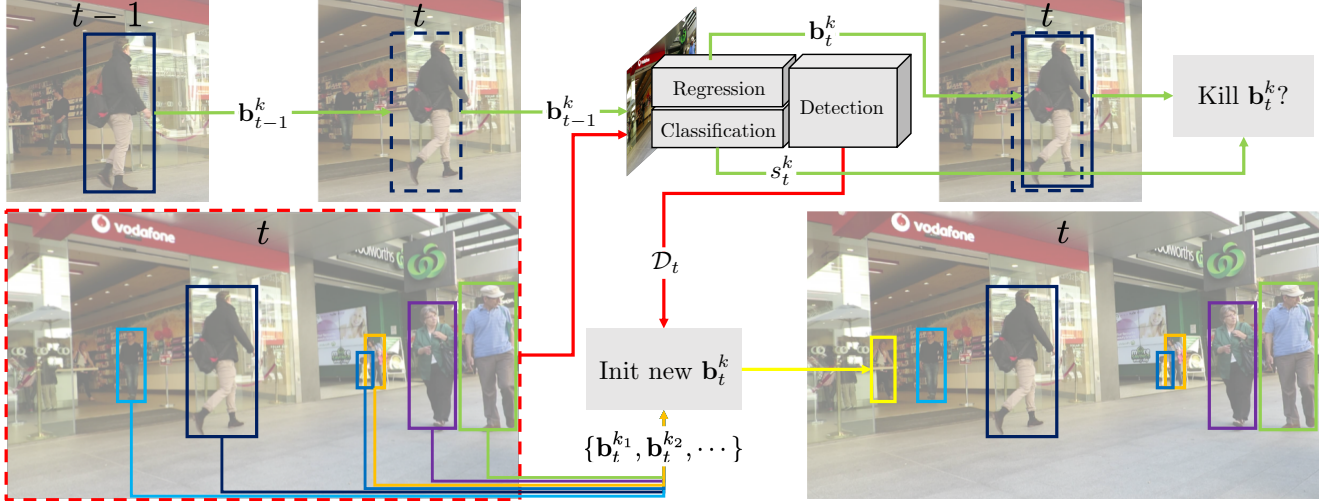


Figure 1: The presented Tracktor tracker accomplishes multi-object tracking only with an object detector and consists of two primary detection processing steps, indicated in green and red. First, the regression of the object detector regresses bounding boxes \mathbf{b}_{t-1}^k of frame $t-1$ to the object’s 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 provides the set of detections \mathcal{D}_t of frame t . Finally, new tracks are initialized if a detection has no Intersection over Union larger than a certain threshold with the set of active tracks $B_t = \{\mathbf{b}_t^{k_1}, \mathbf{b}_t^{k_2}, \dots\}$.

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 multiple object tracking [42], 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 [34, 49, 1]. The family of detectors that evolved to Faster-RCNN [49], and further detectors such as SDP [60], rely on object proposals which are then passed to two heads of a neural network: one for object class classification and the other for regression, which refines the position of the bounding box so that it fits 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 [6, 17, 46] or track-by-track [4]. 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 [5] or, equivalently, minimum cost problem with either fixed costs based on distance [28, 47, 64], including motion models [40], or learned costs [38]. Both formulations can be solved optimally and efficiently. Alternative formulations typically lead to more involved optimization problems, including minimum cliques [63], general-purpose solvers like MCMC [62] or multi-cuts [56]. A recent trend is to design ever more complex models which include other vision input such as reconstruction for multi-camera sequences [41, 57], activity recognition [14], segmentation [45], keypoint trajectories [12] or joint detection [56]. 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 reID. Discriminating pedestrians by appearance is a challenging task and, in fact, it is a problem of its own defined as person re-identification (reID) when the pedestrian has not been tracked for a large period of time. We cannot cover exhaustively the literature on reID but only mention a few works that use appearance models or reID methods to improve multi-object tracking. Color-based appearance models are common [31], but not always reliable, since people can wear very similar clothes, and

color statistics are often contaminated by the background pixels and illumination changes. The authors of [36] borrow ideas from person re-identification and adapt them to “re-identify” targets during tracking. In [59], 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 [37]. Recently, [51] 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 [13, 3], but pedestrian motion gets more complex in crowded scenarios for which researchers have turned to the more expressive Social Force Model [54, 46, 58, 40]. Such a model can also be learned from data [38]. Deep Learning has been extensively used to learn social etiquette in crowded scenarios for trajectory prediction [40, 2, 52]. [65] use single object tracking trained networks to create tracklets for further post-processing into trajectories. Recently, [8, 48] proposed to use reinforcement learning to predict the position of an object in the next frame. While [8] focuses on single object tracking, the authors of [48] train a multi-object pedestrian tracker composed of a bounding box predictor and a decision network for collaborative decision making between tracked objects.

2. A detector is all you need

We propose, in fact, to convert a *detector* into a *Tracker* to perform multiple object tracking. Several CNN-based detection algorithms [49, 60] 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 apply Faster R-CNN [49] with ResNet-101 [23], and re-train it on the MOT17Det [44] 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 [22], 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 to fit it 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 $T_k = \{\mathbf{b}_{t_1}^k, \mathbf{b}_{t_2}^k, \dots\}$, where a bounding box is defined by its coordinates $\mathbf{b}_t^k = (x, y, w, h)$, and t represents a frame of the video. We denote the set object bounding boxes in frame t with $B_t = \{\mathbf{b}_t^{k_1}, \mathbf{b}_t^{k_2}, \dots\}$. 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 $\mathcal{D}_0 = \{\mathbf{d}_0^1, \mathbf{d}_0^2, \dots\} = B_0$. In Figure 1 we illustrate the two subsequent detections processing steps (the nuts and bolts of our method) for all frames $t > 0$ consisting of the bounding box regression and initialization.

Bounding box regression. The first step, denoted with green arrows, exploits the bounding box regression to extend active trajectories to the current frame. This is achieved by regressing the bounding box \mathbf{b}_{t-1}^k of frame $t - 1$ to the object’s new position \mathbf{b}_t^k at frame t . In the case of Faster R-CNN [49], this corresponds to applying the RoI pooling with the previous bounding box coordinates on the features of the current frame. Our assumption is that the target has moved only slightly from one frame to the next, which is usually verified from high frame rates, therefore, the bounding box regressor of the detector is able to snap to the slightly shifted object position. The identity is automatically transferred from the previous to the newly 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 regressed active trajectories \mathbf{b}_t^k is smaller than λ_{new} . That is, we consider a detection as a potentially new object only if it is covering a part of the image 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 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

One might argue that our Tracktor does not really aim at re-detecting the same person in the next frame, but rather re-detecting any close-by person. While true, we show in this section that identity preservation can be greatly enhanced by using two simple extensions: a motion model and a re-identification algorithm. These are commonly used to enhance, e.g., graph-based tracking methods.

Motion model Our previous assumption that the position of an object changes only slightly from frame to frame does not hold in two scenarios: under large camera motion and with videos at low frame rates. In extreme cases, the initial bounding boxes from frame $t - 1$ might not contain the tracked object in frame t at all. We apply two types of simple motion models that will give us better estimations of the position of a target in future frames. For sequences with a moving camera, we apply a simple camera motion compensation (CMC) by aligning frames via image registration using Enhanced Correlation Coefficient (ECC) maximization as introduced in [18]. To this end, we differentiate between two different alignment modes for either a only rotating or also translating camera movement. For sequences with comparatively low frame rates, we apply a constant velocity assumption (CVA) for all objects as in [13, 3].

Re-identification (reID) In order to keep our tracker online, we suggest a short-term reID based on appearance vectors generated by a Siamese neural network [7, 26, 51]. To that end, we store deactivated tracks in their previous non-regressed version \mathbf{b}_{t-1}^k for a fixed number of F_{reID} frames. We then apply our motion model and compare the deactivated with the newly detected tracks. A comparison via the distance in the embedding space is performed by a pre-trained sibling CNN that computes an appearance feature vector for each of the bounding boxes. To minimize the risk of false reIDs, we only consider pairs of deactivated and new bounding boxes with an IoU larger than a fixed threshold for such a comparison.

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. 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 is extremely good at identity preservation.¹

MOTChallenge The multiple object tracking benchmark MOTChallenge² 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* [44], *MOT16* [44] and *MOT17* [44]. The MOT17 [44] 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 [20], Faster R-CNN [49] and SDP [61]. The MOT16 [44] benchmark contains the same sequences as MOT17 but only provides DPM public detections. The 2D MOT 2015 [39] benchmark provides ACF [16] 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) [50] 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 trackers, a trajectory is *only* initialized from a public detection bounding box, i.e. we *never* use our Faster R-CNN to detect a new bounding box. Only the bounding box regressor and computation of the classification score s_t^k are used. Note, there are several methods on the MOTChallenge benchmark that also use a trained neural network to score detections [30, 10, 15], therefore, we consider the comparison with public methods as fair. It is worth noting that our trained Faster R-CNN yields a detection performance on par with the Faster R-CNN entry on the official MOT17Det detection benchmark.

3.1. Ablation study

To demonstrate the potential performance improvements from extending our vanilla Tracktor tracker with tracking specific extensions, we perform an ablation study on the MOT17 [44] training set. In Table 1, we show the results of introducing re-identification (reID) and camera motion compensation (CMC). Despite the simple nature of these extensions, their contribution is significant towards the drastic reduction of identity switches. Note the increment of 3

¹Tracktor code: https://github.com/coming_soon.

²The MOTChallenge web page: <https://motchallenge.net>.

Method	MOTA \uparrow	IDF1 \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	ID Sw. \downarrow
Tracktor	56.7	59.2	29.2	25.2	7158	133284	5286
Tracktor+reID	57.3	60.6	29.5	25.2	7158	133284	3469
Tracktor+CMC	58.3	61.3	30.6	25.3	7017	131655	1927
Tracktor++ (reID + CMC)	58.8	62.3	30.6	25.3	7017	131655	1425

Table 1: This ablation study illustrates the incremental improvements of our vanilla Tracktor extended by 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 on all three sets of MOT17 public detections. The arrows indicate low or high optimal metric values.

Method	MOTA \uparrow	IDF1 \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	ID Sw. \downarrow
MOT17 [44]							
Tracktor++	52.4	52.4	18.6	37.2	19025	246768	2816
FWT [24]	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 [10]	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 [44]							
Tracktor++	53.6	52.8	19.0	36.6	5217	78471	909
LMP [55]	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 [24]	47.8	44.3	19.1	38.2	8886	85487	852
MOTDT [10]	47.6	50.9	15.2	38.3	9253	85431	792
2D MOT 2015 [39]							
Tracktor++	42.9	47.9	25.4	25.2	8338	25813	926
AP_HWDPL_p [9]	38.5	47.1	8.7	37.4	4005	33203	586
AMIR15 [53]	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 [19]	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.

percentage points in the IDF1 measure. As expected, the identity preservation capabilities of the detector greatly benefit from these extensions. In the next section, we show that this effect successfully translates to a comparison with other state-of-the-art methods.

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. A detailed summary of our results on individual sequences will be provided in the supplementary material. For all sequences, camera motion compensation (CMC) and reID

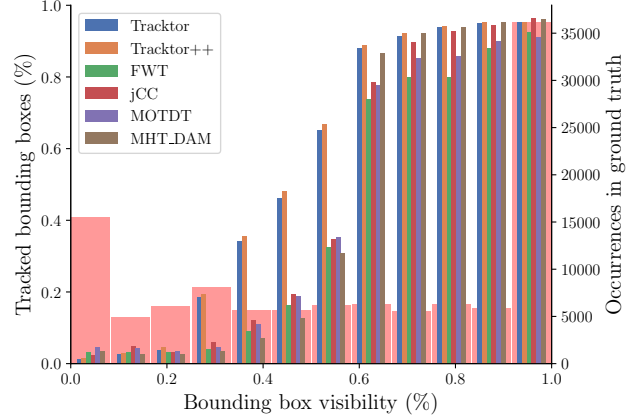


Figure 2: We illustrate the ratio of tracked bounding boxes with respect to their visibility evaluated on the Faster R-CNN public detections. The results clearly demonstrate that none of the presented more sophisticated methods achieves superior performance to our approach. This is especially noticeable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

are used. The only low frame rate sequence is the 2D MOT 2015 *AVG-TownCentre*, for which we apply the mentioned constant velocity assumption (CVA). For the two autonomous driving sequences, originally from KITTI [21] benchmark, we apply the rotation as well as translation camera motion compensation. Note, we use the same Tracktor++ tracker, only 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.

4. Analysis

The superior performance of our tracker despite its simplicity and lack of any tracking specific training or optimization demands a more thorough analysis. From the nature of Tracktor, it is not expected to excel in crowded and occluded, but rather only in benevolent, tracking scenarios. Which begs the question whether more elaborate tracking methods fail to specifically address these complex scenar-

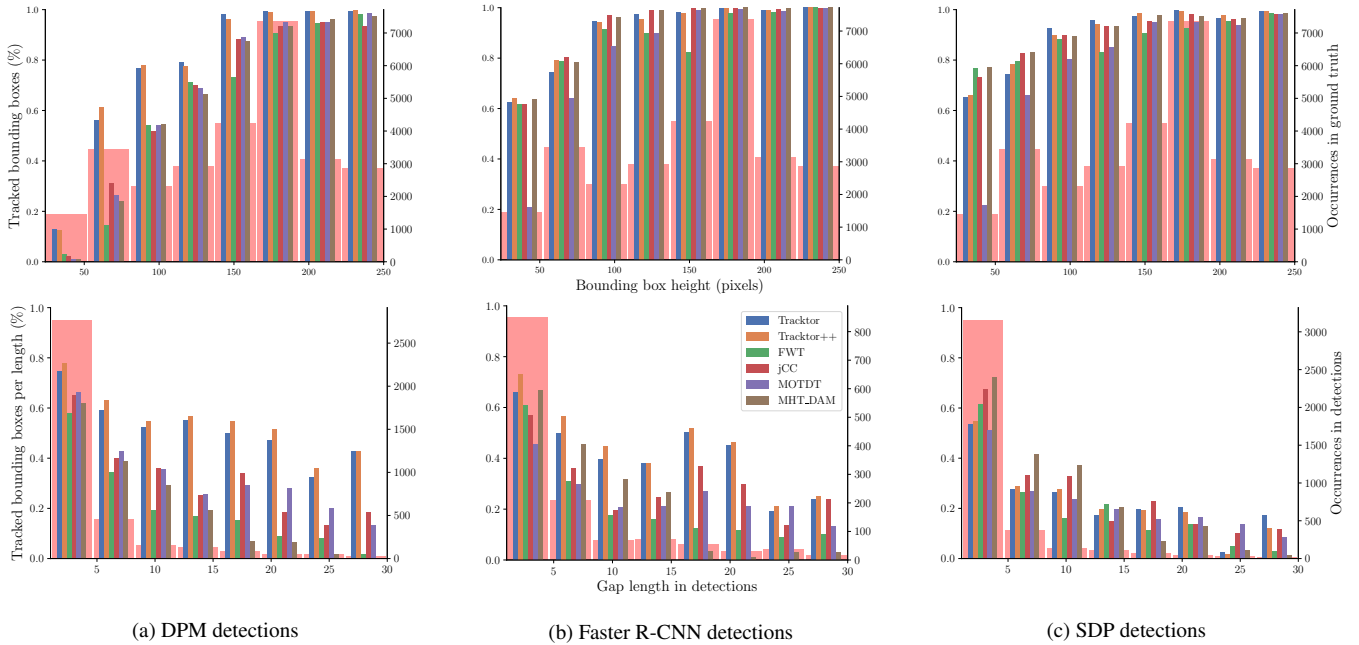


Figure 3: The two rows illustrate the ratio of tracked bounding boxes with respect to (i) bounding box heights and (ii) the length of gaps in the provided public detections. The transparent red bars indicate the ground truth distribution of heights and gap length distribution in the detections, respectively. To demonstrate the shortcomings of the presented trackers we limited the height comparison to bounding boxes with visibility greater or equal than 0.9. Objects that are not detect at all are not considered as detection gap. Hence, SDP which provides overall the most detections also generates the most gaps.

Method	Online	Graph	reID	Appearance model	Motion model	Other
Tracktor						
Tracktor++	×		×		Camera	
FWT [24]		Dense				Face detection
jCC [30]		Dense				Point trajectories
MOTDT17 [10]	×		×	×	Kalman	
MHT_DAM [32]		Sparse		×		

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

ios as well. Our experiments and the subsequent analysis ought to demonstrate the strengths of our approach for easy tracking scenarios and motivate future dedicated tracking methods 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.

4.1. Tracking challenges

In order to understand the performance and behavior of our tracker we want to highlight its strengths and weaknesses and compare them with other trackers on challenging tracking scenarios. To this end, we summarize their fundamental characteristics in Table 3. FWT [24] and jCC [30] both apply a dense offline graph optimization on all detec-

tions in a given sequence. In contrast, MHT_DAM [32] limits its optimization to a sparse forward view of hypothetical trajectories. For all experiments, we use the training set of the MOT17 challenge, and compare to all top performing methods that share this data publicly.

Object visibility Intuitively, we expect our approach to suffer when the bounding box regression encounters either object-object or object-non-object occlusions, in other words, for targets with diminished visibility. In Figure 2, we compare multiple trackers and their ratio of successfully tracked bounding boxes with respect to their visibility. All methods use the provided Faster R-CNN detections. The transparent red curve shows the occurrences of ground truth bounding box for each visibility, and illustrates the proportionate impact of a visibility on the overall performance of the trackers. While all methods perform similar for targets with a visibility larger than 0.5, our method achieves superior performance for mostly occluded bounding boxes. Neither the appearance models of MHT_DAM nor the re-identification of MOTDT17 [10] seem to tackle such complex tracking scenarios. The high MOTA of all presented methods is largely due to the unbalanced distribution of ground truth visibilities. Despite their offline interpolation capabilities, jCC and MHT_DAM do not perform notably better on highly occluded targets. Without a more

sophisticated appearance or motion model the extensions only achieve minor improvements over our vanilla Tracktor.

Object size Interestingly about 40% of the highly visible bounding boxes are not tracked by any method. We argue that in addition to its visibility, the size, specifically height, of an object plays a decisive role in its “trackability”. For tracking-by-detection methods, this depends foremost on the object size robustness of the underlying detection method. We therefore conduct the same comparison of tracking ratios, this time per bounding box height. The plot for the three available public detection sets of MOT17 can be found in Figure 3. To demonstrate the shortcomings even for highly visible bounding boxes, the first row of Figure 3 only shows tracking ratios for bounding boxes with a visibility larger than 0.9. In addition, we restrict our analysis to the most common bounding box heights, i.e., smaller than 250 pixels. For larger heights, all trackers analyzed perform similarly. As we can see, most of the non-tracked highly visible boxes are the ones that are too small to be detected. The three graph methods in Table 3 benefit the most from the additional small detections provided by SDP. However, the online MOTDT17 method and its learned appearance model and reID seems generally vulnerable to small detections. In general, appearance models for small targets are unreliable due to the few observed pixels. Our tracker shows its strength in compensating for insufficient DPM and Faster R-CNN detections for all object sizes. However, we do not see a substantial benefit from the SDP detections. The reason is the classification score of Faster-RCNN, which does not work for small targets. 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 object size analysis already indicated considerable differences in the trackers’ ability to cope with, or benefit from, varying quality of detections. In order to quantify the dependency of SOA methods on the quality of the detection input, we analyze the ability of each tracker to compensate for detection gaps. These are defined as parts of a ground truth trajectory that was at least once detected but is completely covered by detections. In the second row of Figure 3, we compare the tracker’s coverage of each gap vs. the gap length. Intuitively, long gaps should be harder to compensate for, as the tracker is ought perform a strong interpolation. 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 (i) motion prediction with Kalman filter + reID as in MOTDT. With the worse DPM detections, the online method MOTDT does a better job at covering long gaps compared to graph-based jCC. However,

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
Tracktor	56.7	59.2	29.2	25.2	7158	133284	5286
Tracktor++	58.8	62.3	30.6	25.3	7017	131655	1425
Oracle-Kill	58.6	59.0	31.1	24.5	6189	128138	5288
Oracle-REG	61.2	65.6	33.3	24.2	2202	127096	1278
Oracle-MM	59.8	63.2	32.7	23.9	3122	130011	2207
Oracle-reID	57.2	68.2	29.2	25.3	7946	133242	2891
Oracle-MM-reID	59.9	71.8	32.7	24.0	4163	130003	902
Oracle-reID-INTER	56.2	69.5	36.6	22.8	32814	112221	2567
Oracle-MM-reID-INTER	59.5	73.2	41.3	22.2	29570	106136	877
Oracle-ALL	76.7	86.5	53.5	21.2	34	78476	0

Table 4: To show the potential of our Tracktor and focus future research on the most promising tracking challenges, 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 evaluated on all three sets of MOT17 public detections.

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 very long detection gaps. Again, offline methods benefit the most from the improved SDP detections and neither our nor the MOTDT tracker convince with a notable gap length robustness. In our case, this is again largely due to the Faster R-CNN object detector and its inability to regress and score small SDP detections. However, we expect a notable performance gain for the SDP detections in Figure 3c if we replace the Faster R-CNN detector with an SDP object detector.

Identity preservation The extended Tracktor achieves state-of-the-art at identity preservation for two out of the three challenges. For MOT17, only jCC is better. Note, this method is aided by point trajectories, which provide a strong guidance on the motion of the pedestrian, even under partial occlusions. The high IDF1 measure also comes at the expense of having more than 35% more false positives, meaning the tracker creates significantly more ghost trajectories than Tracktor++. The online method MOTDT17 uses reID and in fact we get a similar IDF1 measure, even though we track significantly more pedestrians. Overall, reID measures or other auxiliary inputs, e.g., point trajectories, are key towards good identity preservation, and we have shown that our Tracktor tracker can incorporate such add-ons perfectly.

4.2. Towards a new tracking paradigm

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 two-fold: (i) the impact of the object detector on specific elements such as the killing policy and the bounding box regression, (ii) identify performance upper bounds for potential extensions to our Tracktor. To this end, we create sev-

eral 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.

Oracle detector To illustrate the effect of 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.

In Table 4, we show the results of our Tracktor, its extension and all oracle trackers. While the perfect killing policy does not achieve notable improvements, a detector with perfect regression capabilities would yield substantial performance improvements with respect to MOTA, ID Sw. and FP.

Oracle re-identification and motion model Most notably, our extended Tracktor is already able to compensate for some of the object detector’s insufficiencies with respect to killing and regression. The following oracles ought to illustrate the potential performance gains and upper bounds for our Tracktor extended with perfect reID and motion model. In order to remain online, this excludes any form of hindsight tracking gap interpolation. To this end, we analyse two additional oracles:

- 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 reduce the identity switches substantially. The combined Oracle-MM-reID, which represents the upper bound of our Tracktor++ tracker, promises significant improvements for the IDF1 measure due to its identity preserving characteristics.

Oracle all The omniscient Oracle-ALL performs ground truth killing, regression, reID and motion prediction. The combination of perfect killing and motion model implies that targets will also be tracked during occlusions. We consider its top MOTA of 76.3%, in combination with a high IDF1 and virtually no false positives, as the absolute upper bound of our Tracktor on Faster R-CNN public detections in an online setting. This oracle’s performance is only limited by the failure of the underlying object detection method to detect and regress small or occluded targets.

4.3. Discussion

The substantial gap between Oracle-ALL (76.3% MOTA) and Oracle-reID-MM (59.9% MOTA) demonstrates the necessity of a perfect killing policy, or in more

practical terms a motion prediction model that hallucinates the position of an object through long occlusions. Performing a linear interpolation of the bounding boxes as in Oracle-reID-INTER and Oracle-MM-reID-INTER does not yield a similar performance. This is largely due to wrong linear occlusion paths due to particular long detection gaps. We therefore propose two approaches to apply our Tracktor for future research that try to focus on these challenging tracking problems.

Tracktor with extensions Apply our Tracktor in an online fashion to a given set of detections and extend it tracking specific methods. Many training scenarios with large and highly visible objects will be easily covered by our frame to frame bounding box regression. For the remaining scenarios a proper motion predictor that takes into account the layout of the scene and the configuration of objects seems most promising. In addition, such a hallucinating predictor reduces the necessity for advanced killing and re-identification policies.

Tracklet generation By extending the tracking-by-detection paradigm, one could argue for a tracking-by-tracklet approach. Indeed, many algorithms already use tracklets as input [25, 63], as they are richer in information for computing motion or appearance models. Nonetheless, a specific tracking method is usually used to create these tracklets. We advocate a further use of the detector itself, not only to create sparse detections but frame to frame tracklets. In this view, the *detector* would handle the “easy” tracking situations, while we would require a dedicated *tracking* method to handle the hard cases.

In this work, we have formally defined those hard cases, analyzing the situations in which our framework fails. Maybe paradoxically, we also show that other dedicated *tracking* solutions also fail to solve these situations, raising the question whether current methods are focusing on the real challenges in multiple object tracking.

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 will establish a new tracking paradigm, allowing the full use of a detector’s capabilities to resolve most tracking scenarios, and leaving the hard cases for researchers to focus on.

References

- [1] J. R. ad A. Farhadi. Yolo9000: Better, faster, stronger. *CVPR*, 2017. 2
- [2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3
- [3] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1272, 2011. 3, 4
- [4] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 744–750, 2006. 2
- [5] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011. 2
- [6] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. van Gool. Robust tracking-by-detection using a detector confidence particle filter. *IEEE International Conference on Computer Vision (ICCV)*, pages 1515–1522, 2009. 2
- [7] J. Bromley, I. Guyon, Y. LeCun, E. Saeckinger, and R. Shah. Signature verification using a “siamese” time delay neural network. *NIPS*, 1993. 4
- [8] B. Chen, D. Wang, P. Li, S. Wang, and H. Lu. Real-time ‘actor-critic’ tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3
- [9] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai. Online multi-object tracking with convolutional neural networks. pages 645–649, Sept 2017. 5
- [10] L. Chen, H. Ai, Z. Zhuang, and C. Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification, 07 2018. 4, 5, 6
- [11] X. Chen and A. Gupta. An implementation of faster RCNN with study for region sampling. *CoRR*, abs/1702.02138, 2017.
- [12] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. *ICCV*, 2015. 2
- [13] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. *European Conference on Computer Vision (ECCV)*, pages 553–567, 2010. 3, 4
- [14] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. *European Conference on Computer Vision (ECCV)*, pages 215–230, 2012. 2
- [15] Y. chul Yoon, A. Boragule, Y. min Song, K. Yoon, and M. Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *AVSS*, 2018. 4
- [16] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *PAMI*, 36(8):1532–1545, Aug. 2014. 4
- [17] 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
- [18] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *PAMI*, 30(10):1858–1865, 2008. 4
- [19] K. Fang, Y. Xiang, and S. Savarese. Recurrent autoregressive networks for online multi-object tracking. *WACV*, abs/1711.02741, 2017. 5
- [20] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *pami*, 32:1627–1645, 2009. 4
- [21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5
- [22] R. B. Girshick. Fast r-cnn. *ICCV*, pages 1440–1448, 2015. 3
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, abs/1512.03385, 2015. 3
- [24] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. *CVPR*, abs/1705.08314, 2017. 5, 6
- [25] R. Henschel, L. Leal-Taixé, and B. Rosenhahn. Efficient multiple people tracking using minimum cost arborescences. *German Conference on Pattern Recognition (GCPR)*, 2014. 8
- [26] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 4
- [27] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CVPR*, abs/1611.10012, 2016.
- [28] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2
- [29] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. 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] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *PAMI*, pages 1–1, 2018. 4, 5, 6
- [31] C. Kim, F. Li, A. Ciptadi, and J. Rehg. Multiple hypothesis tracking revisited: Blending in modern appearance model. *ICCV*, 2015. 2
- [32] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4696–4704, Dec 2015. 5, 6
- [33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

- [34] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems (NIPS)*, 2012. 2
- [35] H. W. Kuhn and B. Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, pages 83–97, 1955.
- [36] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? *CVPR*, 2011. 3
- [37] L. Leal-Taixé, C. Canton-Ferrer, and K. 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
- [38] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 3
- [39] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942*, 2015. 4, 5,
- [40] L. Leal-Taixé, G. Pons-Moll, and B. 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
- [41] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2
- [42] W. Luo, X. Zhao, and T.-K. Kim. Multiple object tracking: A review. *arXiv:1409.7618 [cs]*, Sept. 2014. 2
- [43] C. Ma, C. Yang, F. Yang, Y. Zhuang, Z. Zhang, H. Jia, and X. Xie. Trajectory factory: Tracklet cleaving and reconnection by deep siamese bi-gru for multiple object tracking. *ICME*, abs/1804.04555, 2018. 5
- [44] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 1, 3, 4, 5,
- [45] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [46] S. Pellegrini, A. Ess, K. Schindler, and L. 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
- [47] H. Pirsiavash, D. Ramanan, and 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
- [48] L. Ren, J. Lu, Z. Wang, Q. Tian, and J. Zhou. Collaborative deep reinforcement learning for multi-object tracking. *ECCV*, 2018. 3
- [49] S. Ren, R. G. K. He, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NIPS)*, 2015. 1, 2, 3, 4,
- [50] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. *ECCV Workshops*, 2016. 4,
- [51] E. Ristani and C. Tommasi. Features for multi-target multi-camera tracking and re-identification. *CVPR*, 2018. 3, 4
- [52] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction. *European Conference on Computer Vision (ECCV)*, 2016. 3
- [53] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *ICCV*, abs/1701.01909, 2017. 5
- [54] P. Scovanner and M. Tappen. Learning pedestrian dynamics from the real world. *IEEE International Conference on Computer Vision (ICCV)*, pages 381–388, 2009. 3
- [55] S. Tang, M. Andriluka, B. Andres, and B. 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. 5
- [56] S. Tang, M. Andriluka, and B. Schiele. Multi people tracking with lifted multicut and person re-identification. *CVPR*, 2017. 2
- [57] Z. Wu, T. Kunz, and M. 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
- [58] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where are you going? *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352, 2011. 3
- [59] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. *CVPR*, 2012. 3
- [60] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *CVPR*, 2016. 2, 3
- [61] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *CVPR*, pages 2129–2137, 2016. 4,
- [62] Q. Yu, G. Medioni, and I. 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
- [63] A. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. *ECCV*, 2012. 2, 8
- [64] L. Zhang, Y. Li, and R. 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
- [65] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. Yang. Online multi-object tracking with dual matching attention networks. *ECCV*, 2018. 3

Tracking without bells and whistles

Supplementary Material

Philipp Bergmann*

Tim Meinhardt*

Laura Leal-Taixe

Technical University of Munich

Abstract

The supplementary material complements our work with the pseudocode representation of Tracktor tracker and additional implementation and training details of our object detection method as well as the tracking specific extensions. In addition, we provide more details on our experiments and analysis including the MOTChallenge benchmark results of our Tracktor++ tracker for each sequence and set of public detections.

A. Implementation

In order to complete the presentation of our Tracktor tracker and its extension as well as facilitate a reproduction of our results (code will be published) we provide additional implementation details and references.

A.1. Tracktor

In Algorithm 1 and 2 we present a structured pseudocode representation of our Tracktor tracker for private and public detections, respectively. Algorithm 1 corresponds to the method illustrated in Figure 1 and Section 2.2 of our main work.

Object detector As mentioned before, the our approach requires no dedicated training or optimization on tracking ground truth data and performs tracking only with an object detection method. To this end, we train the Faster R-CNN (FRCNN) [49] multi-object detector on the MOT17Det [44] dataset following the improvements suggested by [11]. These include a replacement of the Region of Interest (RoI) pooling [22] by the *crop and resize* pooling suggested by Huang et al. [27] and training with a batch size of $N = 1$ instead of $N = 2$ while increasing the number of extracted regions from $R = 128$ to $R = 256$. Both changes ought to improve the detection results for comparatively small objects. For additional hints and details we refer to [11]. We

achieve the best results with a ResNet-101 [23] as the underlying feature extractor. In Table 1 we compare the performance of our implementation on the official MOT17Det detection benchmark for the three object detection methods mentioned in this work. The results demonstrate the incremental gain in detection performance of DPM [20], FRCNN and SDP [61] (ascending order). Our FRCNN implementation is on par with the official MOT17Det entry. However, due to the changes suggested in [11] we achieve less false positives and negatives on mostly small targets.

Method	AP \uparrow	MODA \uparrow	FP \downarrow	FN \downarrow	Precision \uparrow	Recall \uparrow
Ours (FRCNN)	0.72	71.6	8227	24269	91.6	78.8
DPM [20]	0.61	31.2	42308	36557	64.8	68.1
FRCNN [49]	0.72	68.5	10081	25963	89.8	77.3
SDP [61]	0.81	76.9	7599	18865	92.6	83.5

Table 1: A comparison of our Faster R-CNN (FRCNN) implementation on the MOT17Det detection benchmark for the three object detection methods mentioned in this work. Our results are on par with the official FRCNN implementation. For a detailed summary of the shown detection metrics we refer to the official MOTChallenge web page <https://motchallenge.net>.

A.2. Tracking extensions

Our presented Tracktor++ tracker is an extension of the Tracktor that uses two multi-pedestrian tracking specific extensions, namely, a motion model and re-identification.

Motion model For the motion model via camera motion compensation (CMC) we apply image registration using the Enhanced Correlation Coefficient (ECC) maximization as in [18]. The underlying image registration allows either for an euclidean or affine image alignment mode. We apply the first for rotating camera movements, e.g., as a result of an unsteady camera movement. In the case of an additional camera translation such as in the autonomous driving sequences of 2D MOT 2015 [44], we resort to the affine trans-

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

formation. It should be noted that in MOT17 [44], camera translation is comparatively slow and therefore we consider all sequences as only rotating. In addition, we present a second motion model which aims at facilitating the regression for sequences with low frame rates, i.e., large object displacements between frames. Before we perform bounding box regression, the constant velocity assumption (CVM) model shifts bounding boxes in the direction of their previous velocity. This is achieved by moving the center of the bounding box \mathbf{b}_{t-1}^k by the vectorial difference of the two previous bounding box centers at $t - 2$ and $t - 1$.

Re-identification Our short-term re-identification utilizes a Siamese neural network to compare bounding box features and return a measure of their identity. To this end, we train the TriNet [26] architecture which is based on ResNet-50 [23] with the triplet loss and *batch hard* strategy as presented in [26]. The network is optimized with Adam [33] with $\beta = (0.9, 0.999)$ and a decaying learning rate as described by Hermans et al. [26]. Training samples with corresponding identity are generated from the MOT17 tracking ground truth training data. The TriNet architecture requires input data with a dimension of $H \times W = 256 \times 128$. To allow for a subsequent data augmentation via horizontal flip and random cropping, each ground truth bounding box is cropped and resized to $\frac{9}{8}(H \times W)$. A training batch consists of 18 randomly selected identities, each of which is represented with 4 different samples. Identities with less than 4 samples in the ground truth data are discarded.

B. Experiments

A detailed summary of our official and published MOTChallenge benchmark results for our Tracktor++ tracker is presented in Table 2. For the corresponding results for each sequence and set of detections for the other trackers mentioned in this work we refer to the official MOTChallenge web page available at <https://motchallenge.net>.

B.1. Evaluation metrics

In order to measure the performance of a tracker we mentioned the Multiple Object Tracking Accuracy (MOTA) [29] and ID F1 Score (IDF1) [50]. However, previous Tables such as 2 included additional informative metrics. The false positives (FP) and negatives (FN) account for the total number of either bounding boxes not covering any ground truth bounding box or ground truth bounding boxes not covered by any bounding box, respectively. To measure the track identity preserving capabilities, we report the total number of identity switches (ID Sw.), i.e., a bounding box covering a ground truth bounding box from a different track than in the previous frame. The mostly tracked (MT) and mostly

lost (ML) metrics provide track wise information on how many ground truth tracks are covered by bounding boxes for either at least 80% or at most 20%, respectively. MOTA and IDF1 are meaningful combinations of the aforementioned basic metrics. All metrics were computed using the official evaluation code provided by the MOTChallenge benchmark.

B.2. Raw DPM detections

As most object detection methods, DPM applies a final non-maximum-suppression (NMS) step to a large set of raw detections. The MOT16 [44] benchmark provides both, the set before and after the NMS, as public DPM detections. However, this NMS step is performed with DPM classification scores and an unknown Intersection over Union (IoU) threshold. Therefore, we extracted our own classification scores for all raw detections and applied our own NMS step. Although not specifically provided, we followed the convention to also process raw DPM detections for MOT17. Note, several other public trackers already work on raw detections [30, 10, 15] and their own classification score and NMS procedure. Therefore, we consider the comparison with public trackers as fair.

B.3. Tracktor thresholds

To demonstrate the robustness of our tracker with respect to the classification score and IoU thresholds, we refrained from any sequence or detection-specific fine-tuning. In particular, we performed our experiments on all benchmarks with $\sigma_{active} = 0.5$, $\lambda_{active} = 0.6$ and $\lambda_{new} = 0.3$, which were chosen to be optimal for the MOT17 training dataset. In general, a higher λ_{active} than λ_{new} introduces stability into the tracker, as less active tracks are killed by the NMS and less new tracks are initialized. A comparatively higher λ_{active} relaxes potential object-object occlusions and implies a certain confidence in the regression performance.

C. Oracle trackers

In our main work, we conclude the analysis in Section 4 with a comparison of multiple oracle trackers that highlight the potential of future research directions. For each oracle, one or multiple aspects of our Tracktor++ tracker are substituted with ground truth information, thereby simulating perfect behavior. For further understanding, we provide more details on the oracles for each of the distinct tracking aspects:

- **Oracle-Kill:** This oracle kills tracks only if they have an IoU less than 0.5 with the corresponding ground truth bounding box. The matching between predicted and ground truth tracks is performed with the Hungarian [35] algorithm. In the case of an object-object occlusion ($\text{IoU} > 0.8$) the ground truth matching is ap-

plied to decide which of the objects is occluded by the other and therefore should be killed.

- **Oracle-REG:** We simulate a perfect regression by matching tracks with an IoU threshold of 0.5 to the ground truth at frame $t - 1$. The next track bounding box is then the corresponding ground truth boxes at t .
- **Oracle-MM:** A perfect motion model works analogous to Oracle-REG but we move the previous bounding box center to the center of the ground truth bounding box at frame t . However, the bounding box height and width are still determined by the regression.
- **Oracle-reID:** In addition to the active tracks we also use the Hungarian algorithm to match the new set of detections to the ground truth data. Possible identity matches between tracks and detections yield a perfect re-identification.

Algorithm 1: Tracktor algorithm (private detections)

Data: Video sequence as ordered list

$I = \{i_0, i_1, \dots, i_{T-1}\}$ of images i_t .

Result: Set of object trajectories

$\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ with
 $T_k = \{\mathbf{b}_{t_1}^k, \mathbf{b}_{t_2}^k, \dots, \mathbf{b}_{t_N}^k \mid 0 \leq t_1, \dots, t_N \leq T-1\}$ as a list of ordered object bounding boxes $\mathbf{b}_t^k = (x, y, w, h)$.

```
1  $\mathcal{T}, \mathcal{T}_{\text{active}} \leftarrow \emptyset;$ 
2 for  $i_t \in I$  do
3    $B, S \leftarrow \emptyset;$ 
4   for  $T_k \in \mathcal{T}_{\text{active}}$  do
5      $\mathbf{b}_{t-1}^k \leftarrow T_k[-1];$ 
6      $\mathbf{b}_t^k, s_t^k \leftarrow \text{detector.reg\_and\_class}(\mathbf{b}_{t-1}^k);$ 
7     if  $s_t^k < \sigma_{\text{active}}$  then
8        $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\};$ 
9        $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\};$ 
10    else
11       $B \leftarrow B + \{\mathbf{b}_t^k\};$ 
12       $S \leftarrow S + \{s_t^k\};$ 
13   $B \leftarrow \text{NMS}(B, S, \lambda_{\text{active}});$ 
14  for  $k, T_k \in \mathcal{T}_{\text{active}}$  do
15    if  $k \notin B$  then
16       $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\};$ 
17       $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\};$ 
18  for  $T_k, \mathbf{b}_t^k \in \text{zip}(\mathcal{T}_{\text{active}}, B)$  do
19     $T_k \leftarrow T_k + \{\mathbf{b}_t^k\};$ 
20   $\mathcal{D}_t \leftarrow \text{detector.detections}(i_t);$ 
21  for  $d_t \in \mathcal{D}_t$  do
22    for  $\mathbf{b}_t^k \in B$  do
23      if  $\text{IoU}(d_t, \mathbf{b}_t^k) > \lambda_{\text{new}}$  then
24         $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{d_t\};$ 
25  for  $d_t \in \mathcal{D}_t$  do
26     $T_k \leftarrow \emptyset;$ 
27     $T_k \leftarrow T_k + \{d_t\};$ 
28     $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} + \{T_k\};$ 
29  $\mathcal{T} \leftarrow \mathcal{T} + \mathcal{T}_{\text{active}};$ 
```

Algorithm 2: Tracktor algorithm (public detections)

Data: Video sequence as ordered list

$I = \{i_0, i_1, \dots, i_{T-1}\}$ of images i_t and public detections as ordered list

$\mathcal{D} = \{\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_{T-1}\}$ of detections \mathcal{D}_t .

Result: Set of object trajectories

$\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ with
 $T_k = \{\mathbf{b}_{t_1}^k, \mathbf{b}_{t_2}^k, \dots, \mathbf{b}_{t_N}^k \mid 0 \leq t_1, \dots, t_N \leq T-1\}$ as a list of ordered object bounding boxes $\mathbf{b}_t^k = (x, y, w, h)$.

```
1  $\mathcal{T}, \mathcal{T}_{\text{active}} \leftarrow \emptyset;$ 
2 for  $i_t, \mathcal{D}_t \in \text{zip}(I, \mathcal{D})$  do
3    $B, S \leftarrow \emptyset;$ 
4   for  $T_k \in \mathcal{T}_{\text{active}}$  do
5      $\mathbf{b}_{t-1}^k \leftarrow T_k[-1];$ 
6      $\mathbf{b}_t^k, s_t^k \leftarrow \text{detector.reg\_and\_class}(\mathbf{b}_{t-1}^k);$ 
7     if  $s_t^k < \sigma_{\text{active}}$  then
8        $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\};$ 
9        $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\};$ 
10    else
11       $B \leftarrow B + \{\mathbf{b}_t^k\};$ 
12       $S \leftarrow S + \{s_t^k\};$ 
13   $B \leftarrow \text{NMS}(B, S, \lambda_{\text{active}});$ 
14  for  $k, T_k \in \mathcal{T}_{\text{active}}$  do
15    if  $k \notin B$  then
16       $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\};$ 
17       $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\};$ 
18  for  $T_k, \mathbf{b}_t^k \in \text{zip}(\mathcal{T}_{\text{active}}, B)$  do
19     $T_k \leftarrow T_k + \{\mathbf{b}_t^k\};$ 
20   $S \leftarrow \emptyset;$ 
21  for  $\mathbf{d}_t \in \mathcal{D}_t$  do
22     $\mathbf{d}_t, s_t \leftarrow \text{detector.reg\_and\_class}(\mathbf{d}_t);$ 
23    if  $s_t < \sigma_{\text{active}}$  then
24       $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{\mathbf{d}_t\};$ 
25    else
26       $S \leftarrow S + \{s_t\};$ 
27   $\mathcal{D}_t \leftarrow \text{NMS}(\mathcal{D}_t, S, \lambda_{\text{new}});$ 
28  for  $\mathbf{d}_t \in \mathcal{D}_t$  do
29    for  $\mathbf{b}_t^k \in B$  do
30      if  $\text{IoU}(\mathbf{d}_t, \mathbf{b}_t^k) > \lambda_{\text{new}}$  then
31         $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{\mathbf{d}_t\};$ 
32  for  $d_t \in \mathcal{D}_t$  do
33     $T_k \leftarrow \emptyset;$ 
34     $T_k \leftarrow T_k + \{d_t\};$ 
35     $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} + \{T_k\};$ 
36  $\mathcal{T} \leftarrow \mathcal{T} + \mathcal{T}_{\text{active}};$ 
```

Sequence	Detection	MOTA \uparrow	IDF1 \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	ID Sw. \downarrow
MOT17 [44]								
MOT17-01	DPM [20]	39.9	38.1	20.8	50.0	133	3712	33
MOT17-03	DPM	66.5	59.8	34.5	12.2	1405	33420	208
MOT17-06	DPM	52.3	53.3	18.9	39.6	242	5279	103
MOT17-07	DPM	30.5	36.4	6.7	43.3	1474	10085	179
MOT17-08	DPM	27.5	29.8	10.5	48.7	204	15026	89
MOT17-12	DPM	45.4	55.3	17.6	47.3	73	4626	31
MOT17-14	DPM	23.0	30.7	6.1	53.7	977	13045	215
MOT17-01	FRCNN [49]	36.9	38.3	20.8	41.7	472	3567	34
MOT17-03	FRCNN	67.1	61.6	34.5	12.8	1650	32636	190
MOT17-06	FRCNN	54.7	55.9	23.4	28.4	432	4785	119
MOT17-07	FRCNN	26.2	36.1	6.7	40.0	2219	10068	175
MOT17-08	FRCNN	27.5	30.8	10.5	50.0	259	14974	86
MOT17-12	FRCNN	42.8	52.8	15.4	48.4	175	4760	22
MOT17-14	FRCNN	20.4	30.3	5.5	53.0	1909	12495	310
MOT17-01	SDP [61]	39.4	38.7	20.8	41.7	362	3503	42
MOT17-03	SDP	70.5	63.0	41.9	10.1	2084	28539	238
MOT17-06	SDP	55.3	56.2	24.8	29.3	438	4716	109
MOT17-07	SDP	27.4	36.1	6.7	40.0	2217	9846	198
MOT17-08	SDP	28.3	30.9	13.2	48.7	329	14718	108
MOT17-12	SDP	45.0	55.7	16.5	47.3	191	4544	33
MOT17-14	SDP	21.6	30.7	4.9	52.4	1780	12424	294
All		52.4	52.4	18.6	37.2	19025	246768	2816
MOT16 [44]								
MOT16-01	DPM	40.2	36.1	26.1	43.5	149	3646	32
MOT16-03	DPM	66.9	60.8	36.5	12.2	1537	32840	219
MOT16-06	DPM	53.4	55.7	21.3	38.5	302	4970	100
MOT16-07	DPM	30.5	35.1	7.4	31.5	1790	9344	207
MOT16-08	DPM	35.1	36.4	12.7	36.5	293	10473	104
MOT16-12	DPM	47.3	55.9	18.6	44.2	95	4247	29
MOT16-14	DPM	23.1	31.1	5.5	53.0	1051	12951	218
All		53.6	52.8	19.0	36.6	5217	78471	909
2D MOT 2015 [39]								
TUD-Crossing	ACF [16]	74.2	50.9	53.8	7.7	16	249	19
PETS09-S2L2	ACF	50.2	30.0	14.3	4.8	486	4010	303
ETH-Jelmoli	ACF	55.3	66.4	35.6	24.4	343	771	19
ETH-Linthescher	ACF	48.3	53.1	13.2	48.7	245	4325	47
ETH-Crossing	ACF	40.8	53.6	11.5	38.5	35	545	14
AVG-TownCentre	ACF	53.0	61.8	40.3	14.2	874	2232	253
ADL-Rundle-1	ACF	13.3	42.1	25.0	18.8	3997	3994	76
ADL-Rundle-3	ACF	45.4	45.6	18.2	18.2	817	4661	73
KITTI-16	ACF	47.1	65.1	17.6	5.9	190	685	25
KITTI-19	ACF	45.4	56.6	17.7	17.7	691	2162	64
Venice-1	ACF	37.4	41.3	23.5	23.5	644	2179	33
All		42.9	47.9	25.4	25.2	8338	25813	926

Table 2: A detailed summary of the tracking results of our Tracktor++ tracker on all three MOTChallenge benchmarks. The results are separated into individual sequences and sets of public detections.