Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling

Daniel Stadler[1,2,3] Jürgen Beyerer[2,1,3]
[1]Karlsruhe Institute of Technology [2]Fraunhofer IOSB [3]Fraunhofer Center for Machine Learning
{daniel.stadler,juergen.beyerer}@iosb.fraunhofer.de

Abstract

Multi-pedestrian trackers perform well when targets are clearly visible making the association task quite easy. However, when heavy occlusions are present, a mechanism to re-identify persons is needed. The common approach is to extract visual features from new detections and compare them with the features of previously found tracks. Since those detections can have substantial overlaps with nearby targets – especially in crowded scenarios – the extracted features are insufficient for a reliable re-identification. In contrast, we propose a novel occlusion handling strategy that explicitly models the relation between occluding and occluded tracks outperforming the feature-based approach, while not depending on a separate re-identification network. Furthermore, we improve the track management of a regression-based method in order to bypass missing detections and to deal with tracks leaving the scene at the border of the image. Finally, we apply our tracker in both temporal directions and merge tracklets belonging to the same target, which further enhances the performance. We demonstrate the effectiveness of our tracking components with ablative experiments and surpass the state-of-the-art methods on the three popular pedestrian tracking benchmarks MOT16, MOT17, and MOT20.

1. Introduction

The tracking of multiple pedestrians is an important component in many different applications, especially surveillance related tasks. The goal is to detect and identify each pedestrian in a video throughout the whole sequence by assigning every target a unique ID.

Most of the existing approaches follow the *tracking-by-detection* paradigm [4, 5, 28, 30, 32, 33, 34] first generating a set of bounding boxes for each frame independently and then associating those detections to tracks based on motion patterns or visual cues. While this association step is easy to perform when all pedestrians are clearly visible, the problem gets much harder in crowded scenes, where occlusions lead to missing detections. To recover pedestrians, many methods extract deep features from the detected bounding boxes with a re-identification model [12, 28, 31, 33, 34]. The problems of these approaches are two-fold. First, besides the object detector, a separate network has to be trained, which is time consuming and increases the computational complexity of the whole tracking pipeline. Second, the image areas of the detected bounding boxes often include parts of nearby pedestrians, especially in crowded scenarios. This harms the representation ability of the extracted visual features and can cause false re-identifications.

In contrast, we propose a new strategy to re-identify targets that explicitly models the occlusion of pedestrians and only takes their motion into account. We introduce the concept of *occluding* and *occluded* tracks and check for new arriving detections, whether they belong to a previously found occluded track only considering its motion, thus removing the need for a separate re-identification network.

The two new tracking states are combined with the idea of *active* and *inactive* tracks with a sophisticated track management that also includes the termination of tracks which are leaving the scene. To utilize the recognized tracks in the consecutive frame, we follow the *tracking-by-regression* paradigm first introduced in [2], where the regression head of a two-stage object detector is exploited to regress the previously found bounding boxes to their new position, making the association step obsolete. We extend this regression to inactive tracks, while preferring active ones in the subsequent non-maximum suppression (NMS) in order to allow a regression-based re-identification without the confusion of active and inactive tracks under occlusion.

Furthermore, we propose an offline extension, which can be applied to any multi-object tracker, that exploits a video sequence in both temporal directions and merges intermediate results to further boost the tracking performance.

The effectiveness of our components is shown with ablative experiments and state-of-the-art results are achieved on the three benchmarks MOT16 [23], MOT17 [23], and MOT20 [8].

To summarize, the main contributions of our work are as follows:

- We propose a new strategy introducing the concepts of *occluding* and *occluded* tracks for re-identifying occluded pedestrians without the need for a separate re-identification network.
- The track management of a regression-based tracker is extended by our occlusion handling, the regression of inactive tracks to bypass missing detections, and the treatment of tracks moving over the image boundary.
- An offline extension is proposed that applies our tracker in both temporal directions and combines the results with a merging mechanism, which significantly improves the overall performance.

2. Related Work

Tracking-by-Detection. The predominant majority of existing methods separate the multiple object tracking (MOT) task into two subproblems – detection and association. As object detection is a large research field itself, many MOT methods focus on the association step, where different cues (position, motion, visual appearance, pose, *etc*.) are considered for linking detections to tracks. SORT [4] propagates the position of tracks to the following frame with a Kalman filter and associates detection boxes to tracks based on overlap measured by Intersection over Union (IoU). The further development DeepSORT [33] integrates deep visual features extracted by a convolutional neural network (CNN) into the association process, while in [30] also human poses are incorporated. The idea of combining different cues in order to get high quality similarity measures is also pursued in many other tracking frameworks [12, 28, 31, 34]. One disadvantage of these approaches lies in the need for designing and training separate networks, which comes with additional computational costs. Contrarily, in [32], an appearance embedding model is trained together with the detection model in a shared network to make the whole MOT system more efficient.

Joint Detection and Tracking. In contrast of performing detection and association one after another, some recent approaches integrate detection and tracking more tightly. D&T [10] uses a single network to learn detection and cross-frame regression of boxes at the same time and aid the training process by introducing correlation features. CenterTrack [39] takes a heatmap of previous track positions as input and predicts, besides the bounding boxes from the underlying CenterNet [40] detector, offset vectors representing the motion of objects. In [24], a 3D CNN is designed to regress bounding boxes for multiple frames simultaneously yielding short tracklets that afterwards are merged to get the final tracks. Tracktor [2] leverages the regression head of a Faster R-CNN [26] detector to regress the previously found tracks in the current frame and, hence, implicitly solves the data association problem. As this *tracking-by-regression* paradigm achieves promising results on pedestrian tracking benchmarks, it has been adopted by some other methods [17, 22, 35] and also serves as baseline for our approach. Since the regression stops when severe occlusion occurs, Tracktor is extended by a re-identification model to re-activate inactive tracks. In contrast, we propose an occlusion handling strategy that does not rely on the extraction of visible features with a separate model. A second difference between our approach and Tracktor lies in the additional regression of inactive tracks that, together with our improved track management, effectively bypasses missing detections and makes the re-activation of inactive tracks possible, even when no new detections arrive.

Tracking through Occlusions. Comparing visual features extracted by a re-identification model is the most popular technique to retrieve missed targets after occlusions [1, 2, 30]. As stated earlier, this comes with an increased computational complexity and often is insufficient because overlapping parts from nearby occluders harm the representational power of extracted features. Another idea to handle occlusions is to apply a hierarchical data association [1, 29], where at first short-term tracklets are generated followed by different tracklet linking strategies. In [13], it is distinguished between inter-object occlusion and obstacle occlusion that are solved with an attention-based appearance model and a scene structure model which aims at segmenting static obstacles in surveillance scenes, respectively. We also account for both occlusion types with our occlusion handling strategy modeling the concept of one pedestrian occluding another one and the regression of inactive tracks that can retrieve occluded pedestrians re-appearing after being occluded by a static obstacle.

Retrieving missed Detections. To further improve the tracking results by re-discovering missed detections, different post-processing techniques exist. For example, the V-IOU tracker [5] applies a single-object tracker like KCF [16] or Medianflow [18] at the start point of a track in the past temporal direction and at the end point of a track in the future temporal direction. In [14], a trajectory filling strategy is proposed that interpolates fragmented tracks taking both pedestrian motion and camera motion into account. The post processing extension of our approach also aims at recovering missing detections but, differently from the previously mentioned methods, we run our tracker completely in both temporal directions by processing the sequences two times – forward and backwards – and merge the two sets of tracklets to obtain the final tracking results.

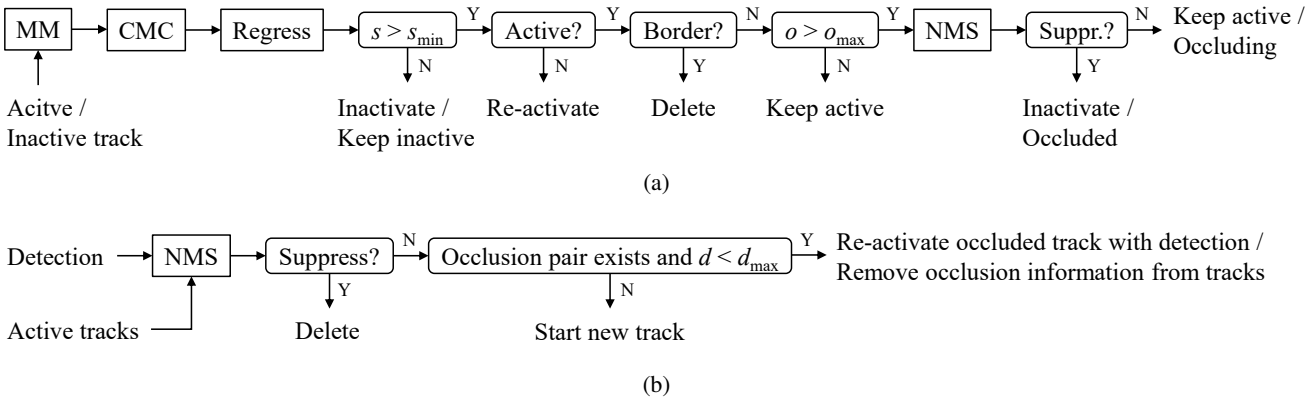


Figure 1: Overview of our track management and occlusion handling. (a) Before the regression, a motion model (MM) and a camera motion compensation (CMC) are applied on both active and inactive tracks. Depending on the regression score s and the overlap with other active tracks o, a track is either inactivated, re-activated, keeps its state, or is deleted in case it moves over the image border. Additionally, if one of two overlapping active tracks gets suppressed by NMS, those tracks are marked as *occluded* and *occluding* as shown in Figure 2. (b) For a new detection, which is not filtered by active tracks via NMS, it is checked with Equation (1), whether the detection belongs to an inactive occluded track. If this is the case, the track is re-activated with the new detection, otherwise a new track is started. For more details refer to Sections 3.1 and 3.2.

3. Proposed Method

Our tracking framework comprises an object detector that is used for regression-based tracking, an occlusion handling strategy, a sophisticated track management, and two motion models for pedestrian and camera motion. A schematic overview is illustrated in Figure 1.

The tracking of multiple pedestrians is realized by exploiting the regression head of a two-stage object detector, as first proposed from Bergmann *et al*. [2] Instead of using a separate network to re-identify occluded persons, we propose an occlusion handling strategy that introduces the concept of *occluding* and *occluded* tracks, while also taking the motion of pedestrians into account. Furthermore, we adopt the standard separation of *active* versus *inactive* tracks but improve the track management by additionally regressing inactive tracks, which enhances the chance of a successful re-identification. Finally, we apply our tracker in both temporal directions and merge the intermediate tracking results to further boost the performance.

3.1. Track Management

Many MOT methods deem a track to be inactive when the regression score falls below a threshold s_{min} (*tracking-by-regression*) or no detection is associated (*tracking-by-detection*) but the reason for this inactivation is not considered. In contrast, we identify four different cases that can make the score s of a regressed box drop:

1. Insufficient quality of image or detector: Motion blur, bad lighting conditions, or other image deficiencies can lead to a low regression score as well as objects that become too small moving away from the camera.

2. Occlusion by objects (vehicles, lanterns, traffic signs, *etc*.): We assume, that the underlying detector of the regression-based tracker does not recognize these objects. Thus, we cannot distinguish this case by item 1.

3. The person is about to leave the camera's field of view.

4. Occlusion by other pedestrians: Different from item 2, other pedestrians are recognized by the regression head of the detector. We take advantage of this in our occlusion handling strategy described in Section 3.2.

As a consequence of the first and the second observation, we propose to also regress inactive tracks. This makes a re-identification possible even when no new detection is available, *i.e*., if the regression score exceeds s_{min}. In the subsequent NMS, that filters strongly overlapping tracks, active tracks are preferred over inactive ones. This is an important detail that prevents an inactive track being regressed to an active track's position which could in turn inactivate the active track. Without preferring active tracks in the NMS, a repeating confusion between active and inactive tracks can occur leading to a large number of identity switches (IDSW) as found in early experiments.

As stated in the third item, when a pedestrian is about to leave the scene at the border of an image, the regression score will drop and the track is inactivated and stays at the image boundary. As we also regress inactive tracks, those

could be mistakenly re-identified by newly entering pedestrians a few frames later. To prevent this, if the regression score s of a track falls below s_{min} and its bounding box is at the border of the image, we calculate the two-dimensional velocity vector of the track and delete it from the set of tracks if the vector points towards the image boundary.

The last item representing the most interesting reason why the regression score can fall bellow s_{min} – the occlusion of pedestrians by other pedestrians – is treated in the following section.

3.2. Occlusion Handling

When two pedestrians cross each other, the overlap of their bounding boxes increases frame by frame until one regressed box is filtered by the NMS because of a too large overlap. Assume track *A* keeps active, while track *B* turns inactive. In this case, we mark track *A* as *occluding B* and track *B* as *occluded by A*. Whenever a new detection arrives and an occlusion track pair exists, the center position of the detection $\mathbf{p}_{\text{D}} = (x_{\text{D}}, y_{\text{D}})$ is compared with the estimated position of the occluded inactive track $\mathbf{p}_{\text{T}} = (x_{\text{T}}, y_{\text{T}})$, after applying a MM and a CMC. If the following inequality holds, the inactive track gets re-activated by the new detection and the occlusion information of the track pair (*occluding B* / *occluded by A*) is deleted.

$$||\mathbf{p}_{\text{D}} - \mathbf{p}_{\text{T}}|| \leq d_{\text{max}} = ||\mathbf{v}_{\text{T}}|| \cdot t_{\text{inactive}} \cdot \alpha \qquad (1)$$

The maximum distance d_{max} for this assignment increases with the velocity of the inactive track $\mathbf{v}_{\text{T}}$ and the number of frames the track has been inactive t_{inactive}, since both terms increase the uncertainty of the estimated position of the inactive track. Moreover, the maximum distance can be tuned by a parameter α. The procedure of our occlusion handling is illustrated in Figure 2. Note that, before the regression of our tracker, we follow a constant velocity assumption in our MM and afterwards apply the CMC from [9] to predict the position of a track in the current frame.

3.3. Tracking in Both Temporal Directions

Whereas our track management and occlusion handling can run online, the extension described in this section aims at generating the best tracking results without the need for an online requirement. Normally, we start a new track whenever a new detection is available and regress it in the *following* frame. However, we also can regress the detection in the *previous* frame to get additional boxes that might have been missed by the detector before. For simplicity, we realize this by applying our tracker two times on each video: One time the video is processed forward and one time backwards in order to get two sets of intermediate tracking results termed *tracklets*. To combine the generated two sets of tracklets, we follow the merging strategy shown in Figure 3.

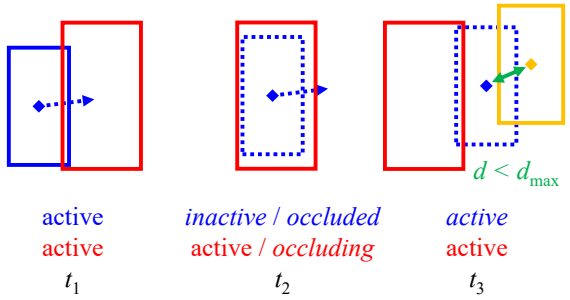


Figure 2: Visualization of our occlusion handling. Two active tracks begin to overlap at t_1 as it happens when two pedestrians cross each other. The blue track of the hidden pedestrian gets inactive at t_2, since the NMS suppresses its regressed box because of a too large overlap and a smaller regression score compared to the red box. The blue track is marked as *occluded* and the red one is marked as *occluding*. The inactive blue track is propagated in consecutive frames with its estimated velocity that is illustrated by the dotted arrow. At t_3, a new detection (yellow box) appears and its center position is compared to the predicted position of the blue track. As the distance d is smaller than d_{max} referring to Equation (1), the blue track is re-activated by the orange detection. A change of state is highlighted with *italic* text.

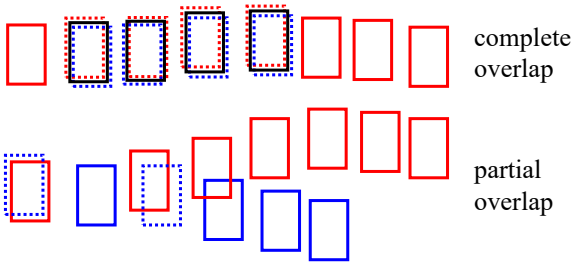


Figure 3: Proposed tracklet merging strategy. Completely overlapping tracklets are merged by averaging their predictions yielding the black boxes. For partially overlapping ones, the highly overlapping boxes of the shorter tracklet are deleted. Removed boxes are depicted with dotted lines.

One can think of the red tracklet generated by tracking forward and the blue tracklet by tracking backwards. If two tracklets overlap *completely*, *i.e.*, the IoU threshold is exceeded in each frame where both tracklets have boxes, we are sure that they belong to the same target and merge them by calculating the average boxes shown in black. If the two tracklets are only *partially* overlapping, *i.e.*, there exist frames with no overlap (IoU smaller than the threshold) as it is the case for trajectories of two crossing pedestrians, we delete the boxes of the shorter tracklet in each frame, where the overlap is higher than the threshold, arguing that long tracklets are probably more often correct than short ones.

4. Experiments

4.1. Datasets

We test our approach on the three popular datasets for multiple pedestrian tracking MOT16 [23], MOT17 [23], and MOT20 [8]. The datasets have in common that a public set of detections is provided by the authors in order to allow a fair comparison between different tracking methods. While MOT16 and MOT17 contain the same videos, 7 for training and 7 for testing, the public detections differ: In MOT16, only one set of detections from a deformable part-based model [11] is available, whereas in MOT17 also detections from Faster R-CNN [26] and SDP [36] are given. Since this makes the evaluation of tracking results more independent with respect to the applied detector, we use the train split of MOT17 for all ablative experiments. For MOT20, only one set of detections from Faster R-CNN is provided for a total of 8 videos, 4 for training and 4 for testing. Note that the annotations for the test sequences are not publicly available and the tracking results have to be submitted to the official evaluation server (*motchallenge.net*). All of the three datasets are very challenging including crowded scenes with heavy occlusions, camera motion, and both day and night sequences.

4.2. Regression Model

As underlying model for our regression-based tracker, we choose Faster R-CNN [26] with a FPN [20] as neck and a ResNet-101 [15] as backbone. In our experiments, only the second stage of the Faster R-CNN is used for regressing the public detections of the datasets and no additional detections are generated, *i.e.*, the region proposal network (RPN) is discarded. For simplicity, all images of the train split are utilized in the training and no validation split is used, as the optimization of the regression model is not the focus of our work. Note that our regression-based tracker therefore may perform a bit over-confident in the ablation study because the regression model is trained on the same data. However, the generalization ability of the tracker is proven on the test splits in Section 4.4, where state-of-the-art-results are achieved. Since the sequences in the MOT20 dataset differ quite a lot from the MOT16 / MOT17 ones, we train two separate models on the respective train splits but with the same settings. We use the MMDetection toolbox [7] and initialize the network weights with COCO [21] pre-trained models. Training is performed on the full images using SGD with an initial learning rate of 0.01, batch size of 2, momentum of 0.9, and weight decay of 0.0001.

4.3. Ablation Study

Occlusion Handling versus Re-Identification Model. The main purpose of our occlusion handling (OCC) strategy is to make the use of a separate re-identification network unnecessary. Since our method models the interaction of pedestrians introducing the concept of *occluding* and *occluded* tracks and, therefore, is purely logic-based, there is no need for training a network on large pedestrian datasets and no feature extraction has to be done for the re-identification. We compare the proposed occlusion handling with a re-identification model from [2] (REID) and evaluate our tracker with different *inactive patience*. This is a parameter of the track management determining the successive number of frames a track can remain inactive before it is deleted. Figure 4 shows both the MOTA [3] metric representing the overall tracking performance and the IDF1 [27] metric that focuses on association accuracy.

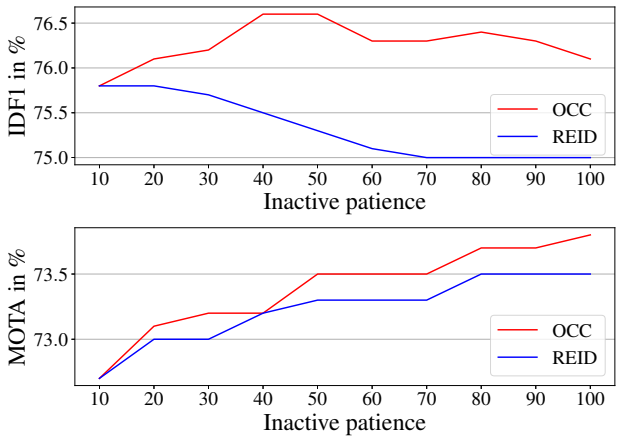
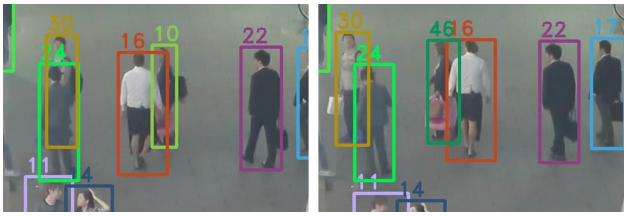


Figure 4: Comparison of our occlusion handling strategy (OCC) with the re-identification baseline (REID).

For short occlusions, the overlaps are mostly not very large and the REID baseline performs on par with OCC. However, for longer occlusions which often contain severe overlaps, the feature computation is harmed, so that the appearance-based REID is clearly outperformed by the proposed occlusion handling strategy. This is reflected by a falling IDF1 score and a rising IDF1 score for REID and OCC, respectively, when the inactive patience is increased. In our method, the best trade-off between MOTA and IDF1 is achieved with an inactive patience of 50 frames which corresponds to roughly 1.7s. For this value, our occlusion handling improves over the REID baseline by 1.3 points in IDF1 and 0.2 points in MOTA without the need for a separate re-identification network. A qualitative example, where our occlusion handling – unlike the REID baseline – successfully recovers a target is given in Figure 5.

Robustness of Occlusion Handling. Referring to Equation (1), the maximum distance d_{max} that defines whether a new detection is assigned to an occluded inactive track can be tuned by the parameter α. As can be seen in the



(a) REID baseline

(b) Occlusion handling

Figure 5: Qualitative comparison between the REID baseline (a) and our proposed occlusion handling strategy (b). While the REID model fails to re-identify the occluded women with ID 10, our occlusion handling is successful.

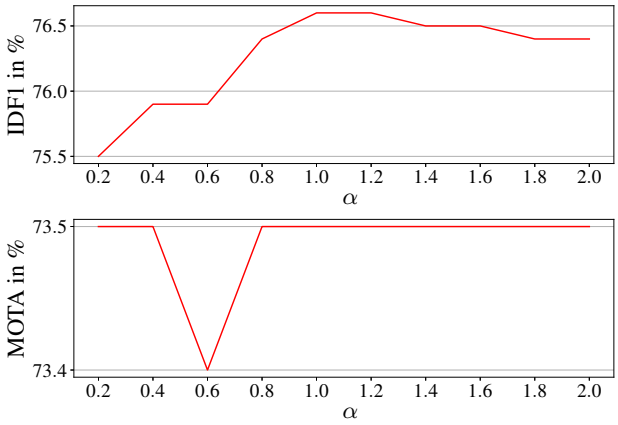


Figure 6: Ablation of the parameter α from Equation (1) of our occlusion handling strategy.

IDF1 metric in Figure 6, the association performance first increases for $\alpha \in [0.2, 1]$ and then slowly decreases for $\alpha > 1.2$. For a larger tolerance α, the number of correct re-identifications increases but also the probability for a wrong assignment gets bigger. The best balance is achieved for $\alpha = 1$. If we double this value, the performance drops only by 0.2 points in IDF1, so we conclude that the occlusion handling is quite robust to the choice of α. This is also reflected in the MOTA score that remains nearly constant (deviation of 0.1 points) for all evaluated values of α.

Table 1: Influence of MM and CMC.

	MOTA	IDF1	FP	FN	IDSW
No motion	71.4	72.4	**2171**	89530	4517
MM	72.2	74.7	2238	88546	2771
CMC	73.3	76.0	2194	87185	696
MM+CMC	**73.5**	**76.6**	2469	**86294**	**648**

MM and CMC. Before the regression of both inactive and active tracks, a MM where we assume a constant velocity of pedestrians is applied. Afterwards, a CMC from [9] aligns the predicted tracks with the current frame. The influence of those components is listed in Table 1. Both models improve the tracking results, whereby the significant enhancement comes from the CMC because sequences with severe camera motion are present in the MOT17 dataset. Combining MM and CMC yields 2.1 points higher MOTA and 4.2 points higher IDF1 compared to the *no motion* baseline. For the CMC, we keep the settings from [2] finding that the parameters are well tuned. In contrast to [2], we additionally apply a MM before the CMC. To predict the position of a track in the next frame, we calculate its velocity by averaging the displacement of the n previous bounding box centers. As illustrated in Figure 7, the IDF1 reaches its maximum for $n = 5$. A too small value makes the velocity estimation noisy, since the regressed bounding boxes are not always perfectly aligned with the pedestrians and the aspect ratio of the boxes can change when pedestrians are moving. However, considering too many past detections for estimating the velocity also introduces errors because the constant velocity assumption is violated. We set $n = 5$ as it yields the best performance in both IDF1 and MOTA.

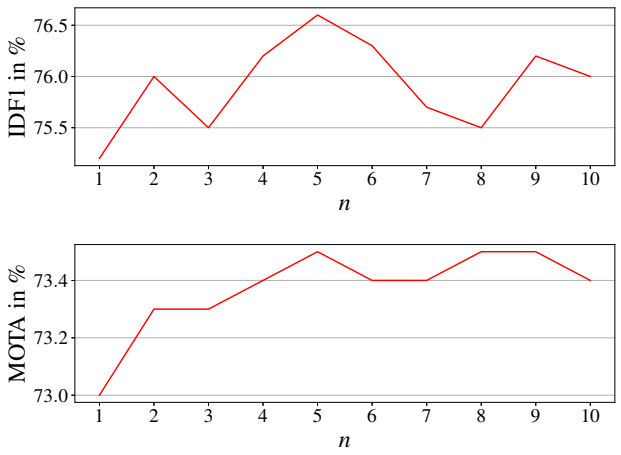


Figure 7: Ablation of the parameter n denoting the number of previous frames that are considered in the velocity estimation of the MM.

Table 2: Impact of our tracking components.

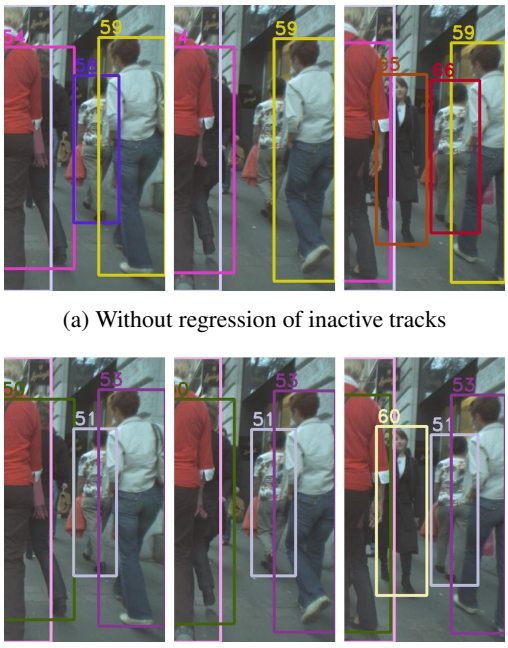
Tracking direction	Regress inactive	Stop border	Occlusion handling	MOTA	IDF1	FP	FN	IDSW
Forward	✗	✗	✗	71.9	74.5	1937	91987	642
Forward	✓	✗	✗	73.4	75.3	2809	86199	704
Forward	✓	✓	✗	73.5	75.5	2503	86222	689
Forward	✓	✓	✓	73.5	76.6	2469	86294	648
Backwards	✗	✗	✗	73.9	75.4	**1664**	85586	**639**
Backwards	✓	✓	✓	75.6	76.8	2350	79183	688
Forw. + Backw.	✓	✓	✓	**80.8**	**78.5**	4227	**58962**	1351

Impact of Our Tracking Components. To get a deeper insight into our method, we run several experiments adding each tracking component one after another. We start from a baseline that includes MM and CMC and regresses only active tracks. After that, we also regress inactive tracks, whereby active tracks are preferred over inactive ones in the NMS step as explained in Section 3.1. Then, we add our border handling to stop tracks at the image boundary leaving the camera's field of view. As third extension, we apply the occlusion handling strategy proposed in Section 3.2. The results of these experiments, where the online version of our tracker is applied, *i.e.*, tracking is only performed in *forward* direction, are shown in the first rows of Table 2.

The regression of inactive tracks substantially reduces the number of false negatives (FN) with a smaller increase in false positives (FP) and IDSW resulting in a gain of 1.5 points in MOTA and a gain of 0.8 points in IDF1. A successful example of the regression of inactive tracks can be seen in Figure 8. Deleting tracks that leave the scene at the image border prevents some false associations for newly entering pedestrians, which yields further improvements. The proposed occlusion handling strategy enhances the IDF1 by another 1.1 points showing its strengths in re-identifying occluded tracks.

Since our offline extension builds upon tracklets generated by tracking in both temporal directions, we additionally apply our tracker *backwards* on the MOT17 train sequences. The results are given in the middle rows of Table 2. One can see, that our tracker with all its components achieves similar improvements over the baseline (+1.7 MOTA / +1.4 IDF1) as when tracking is performed in the standard direction. Another interesting finding is that the MOTA is 2.1 points higher for tracking backwards compared to tracking forward. We hypothesize that the reason for this performance gap lies in the distribution of public detections in the sequences, since missed detections can only be retrieved by the regression-based tracker from that point in the sequence, where the corresponding object is first recognized in terms of public detections.

Finally, we evaluate the performance of our offline extension, where we merge the two sets of tracklets from tracking forward and backwards, respectively, as described in Section 3.3. The results can be perceived in the last row of Table 2. The number of FP and IDSW is increased w.r.t. the online variant, but, at the same time, the number of FN is significantly more reduced raising the MOTA to 80.8% and the IDF1 to 78.5%, underlining the effectiveness of our offline extension. That the approach significantly reduces the number of FN and in total more targets are tracked can be seen in Figure 9.

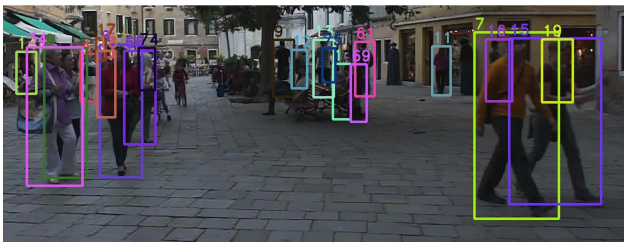


(a) Without regression of inactive tracks

(b) With regression of inactive tracks

Figure 8: Bypassing missing detections with the regression of inactive tracks. (a) Without regressing inactive tracks, the blue track with ID 58 is lost after its regression score falls below s_{min} and cannot be re-identified (ID 66) when a new detection arrives. (b) With the regression of inactive tracks, missing detections are recovered and the same target (ID 51) is successfully tracked.

(a) Tracking only forward



(b) Tracking both forward and backwards

Figure 9: Comparison of tracking only forward (a) and tracking in both temporal directions with subsequent tracklet merging yielding much more tracks (b).

4.4. Comparison with the State-of-the-Art

We compare our tracker, that we term TMOH for *Track Management and Occlusion Handling*, with the state-of-the-art approaches on the test sets of the three multiple pedestrian tracking benchmarks MOT16, MOT17, and MOT20. For a description of these datasets refer to Section 4.1. We use the provided public detection sets for a fair comparison and submit the generated tracks to the official evaluation server. The results for the best officially published and peer reviewed methods on the three benchmarks are listed in Table 3, separated in *online* and *offline* approaches and sorted with ascending MOTA. Note that we evaluated not only our online variant but also the offline extension, where tracking is performed in both temporal directions and the intermediate tracklets are merged to get the final tracks. For both the online and offline entries, our approach surpasses the state-of-the-art on all evaluated benchmarks by a large margin: The second best online (offline) entries regarding the MOTA metric are beaten by 6.2 (5.5) points on MOT16, 0.6 (4.6) points on MOT17, and 7.5 (3.6) points on MOT20. Both the online and offline variants of our tracker also reach the highest IDF1 scores improving over the second best methods by 5.3 (2.3) points on MOT16, 3.2 (0.2) points on MOT17, and 8.5 (3.0) points on MOT20.

5. Conclusion

In this paper, we propose an occlusion handling strategy for a regression-based multi-pedestrian tracker that explicitly models the concept of *occluding* and *occluded* tracks and can successfully retrieve targets without the need for an extra re-identification model. Moreover, we enhance the track management by regressing inactive tracks bypassing missing detections and also cope with tracks leaving the camera's field of view. In addition, we propose to apply our tracker in both temporal directions and merge the intermediate tracklets to get high quality results. In the ablation study, we analyze the impact of our tracking components with extensive experiments. The superiority of our approach is shown on three popular benchmarks, where we achieve state-of-the-art results.

Table 3: State-of-the-art approaches on the test sets of MOT16, MOT17, and MOT20. The entries are categorized in *online* (top rows) and *offline* (bottom rows) methods.

MOT16

Method	MOTA	IDF1	FP	FN	IDSW
PV [19]	50.4	50.8	**2600**	86780	1061
Tracktor++ [2]	54.4	52.5	3280	79149	682
DeepMOT [35]	54.8	53.4	2955	78765	645
GSM [22]	57.0	58.2	4332	73573	475
TMOH (*online*)	63.2	63.5	3122	63376	635
TPM [25]	51.3	47.9	2701	85504	569
MLT [37]	52.8	62.6	5362	80444	**299**
MPNTrack [6]	58.6	61.7	4949	70252	354
Lif_T [17]	61.3	64.7	4844	65401	389
TMOH (*offline*)	**66.8**	**67.0**	5558	**54032**	997

MOT17

Method	MOTA	IDF1	FP	FN	IDSW
Tracktor++ [2]	53.5	52.3	12201	248047	2072
DeepMOT [35]	53.7	53.8	11731	247447	1947
GSM [22]	56.4	57.8	14379	230174	1485
CenterTrack [39]	61.5	59.6	14076	200672	2583
TMOH (*online*)	62.1	62.8	**10951**	201195	1897
MLT [37]	54.8	62.9	19118	234303	**1077**
TT17 [38]	54.9	63.1	20236	233295	1088
MPNTrack [6]	58.8	61.7	17413	213594	1185
Lif_T [17]	60.5	65.6	14966	206619	1189
TMOH (*offline*)	**65.1**	**65.8**	17508	**176389**	2830

MOT20

Method	MOTA	IDF1	FP	FN	IDSW
SORT [4]	42.7	45.1	27521	264694	4470
Tracktor++ [2]	52.6	52.7	**6930**	236680	1648
TMOH (*online*)	60.1	61.2	38043	165899	2342
MLT [37]	48.9	54.6	45660	216803	2187
MPNTrack [6]	57.6	59.1	16953	201384	**1210**
TMOH (*offline*)	**61.2**	**62.1**	54572	**141850**	4154

References

[1] N. M. Al-Shakarji, F. Bunyak, G. Seetharaman, and K. Palaniappan. Multi-object tracking cascade with multi-step data association and occlusion handling. In *IEEE Int. Conf. Adv. Video Sign. Surv.*, 2018.

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *Int. Conf. Comput. Vis.*, pages 941–951, 2019.

[3] K. Bernardin, A. Elbs, and R. Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Eur. Conf. Comput. Vis. Worksh.*, 2006.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE Int. Conf. Image Process.*, pages 3464–3468, 2016.

[5] E. Bochinski, T. Senst, and T. Sikora. Extending iou based multi-object tracking by visual information. In *IEEE Int. Conf. Adv. Video Sign. Surv.*, 2018.

[6] G. Brasó and L. Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6246–6256, 2020.

[7] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[8] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003*, 2020.

[9] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1858–1865, 2008.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *Int. Conf. Comput. Vis.*, pages 3057–3065, 2017.

[11] P. F. Felzenszwalb and D. R. Huttenlocher. Efficient belief propagation for early vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2004.

[12] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. *arXiv:1901.06129*, 2019.

[13] X. Gao and T. Jiang. Osmo: Online specific models for occlusion in multiple object tracking under surveillance scene. In *ACM Int. Conf. Multimedia*, pages 201–210, 2018.

[14] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, X. Pan, and J. Zhao. Mat: Motion-aware multi-object tracking. *arXiv:2009.04794*, 2020.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

[17] A. Hornakova, R. Henschel, B. Rosenhahn, and P. Swoboda. Lifted disjoint paths with application in multiple object tracking. In *Int. Conf. Mach. Learn.*, pages 4364–4375, 2020.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Int. Conf. Pattern Recog.*, pages 2756–2759, 2010.

[19] X. Li, Y. Liu, K. Wang, Y. Yan, and F. Wang. Multi-target tracking with trajectory prediction and re-identification. In *Chin. Auto. Congr.*, pages 5028–5033, 2019.

[20] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 936–944, 2017.

[21] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, pages 740–755, 2014.

[22] Q. Liu, Q. Chu, B. Liu, and N. Yu. Gsm: Graph similarity model for multi-object tracking. In *IJCAI*, pages 530–536, 2020.

[23] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.

[24] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6308–6318, 2020.

[25] J. Peng, T. Wang, W. Lin, J. Wang, J. See, S. Wen, and E. Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, 107:107480, 2020.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[27] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Eur. Conf. Comput. Vis. Worksh.*, pages 17–35, 2016.

[28] S. Sharma, J. A. Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *IEEE Int. Conf. on Rob. Auto.*, pages 3508–3515, 2018.

[29] Y. Song, K. Yoon, Y. Yoon, K. C. Yow, and M. Jeon. Online multi-object tracking with gmphd filter and occlusion group management. *IEEE Access*, 7:165103–165121, 2019.

[30] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3701–3710, 2017.

[31] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *ACM Int. Conf. Multimedia*, page 482–490, 2019.

[32] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang. Towards real-time multi-object tracking. In *Eur. Conf. Comput. Vis.*, pages 107–122, 2020.

[33] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE Int. Conf. Image Process.*, pages 3645–3649, 2017.

[34] J. Xu, Y. Cao, Z. Zhang, and H. Hu. Spatial-temporal relation networks for multi-object tracking. In *Int. Conf. Comput. Vis.*, pages 3987–3997, 2019.

[35] Y. Xu, A. Ošep, Y. Ban, R. Horaud, L. Leal-Taixé, and X. Alameda-Pineda. How to train your deep multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6786–6795, 2020.

[36] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2129–2137, 2016.

[37] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong. Multiplex labeling graph for near-online tracking in crowded scenes. *IEEE Internet Things J.*, 7(9):7892–7902, 2020.

[38] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Lyu, W. Ke, and Z. Xiong. Long-term tracking with deep tracklet association. *IEEE Trans. Image Process.*, 29:6694–6706, 2020.

[39] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In *Eur. Conf. Comput. Vis.*, pages 474–490, 2020.

[40] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019.