High-Speed Tracking-by-Detection Without Using Image Information

Erik Bochinski, Volker Eiselein and Thomas Sikora
Communication System Group, Technische Universität Berlin
Einsteinufer 17, 10587 Berlin
{bochinski, eiselein, sikora}@nue.tu-berlin.de

Abstract

Tracking-by-detection is a common approach to multi-object tracking. With ever increasing performances of object detectors, the basis for a tracker becomes much more reliable. In combination with commonly higher frame rates, this poses a shift in the challenges for a successful tracker. That shift enables the deployment of much simpler tracking algorithms which can compete with more sophisticated approaches at a fraction of the computational cost. We present such an algorithm and show with thorough experiments its potential using a wide range of object detectors. The proposed method can easily run at 100K fps while outperforming the state-of-the-art on the DETRAC vehicle tracking dataset.

1. Introduction

Object tracking is a key technology to semantic video interpretation. As a classical computer vision problem, it gives important information cues to analytics systems such as traffic analysis, sports or forensics. It can also help in reducing the search space for further applications e.g. automatic number plate recognition (ANPR) or face recognition which are both common use cases in the surveillance domain. Multi-object tracking in general scenarios requires both the estimation of an unknown number of objects of interest in a video and their respective paths.

This is especially important in the popular field of tracking-by-detection where first an object detector is applied to each video frame. In a second step, a tracker is used to associate these detections to tracks. A typical challenge for tracking-by-detection systems, especially when applied on-line, has always been the limited performance of the underlying detector which may produce false positive and missed detections. A good tracker should be able to handle these flaws by filling the ”gaps” of missing detections and ignoring false positives. More problems arise when multiple objects cross each other and their paths become ambiguous. Many methods have been proposed to solve these problems: [1, 2] define a continuous energy function and search for strong local minima using sophisticated minimization techniques. [6] estimates short tracklets for unambiguous frames and stitches them according to a dynamics-based similarity. Other approaches include using a globally optimal and locally greedy method and integer linear programming [12] and online discriminative appearance learning [3].

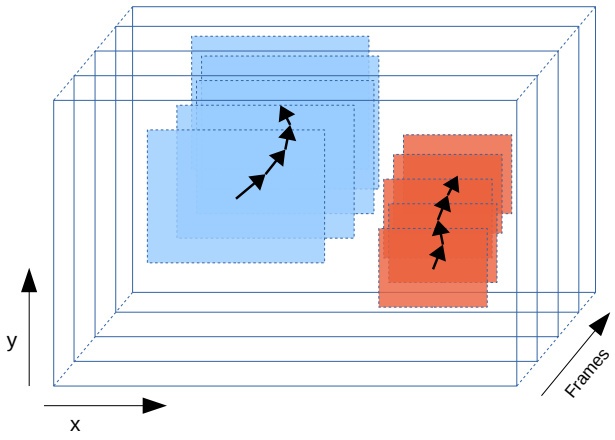


Figure 1. Basic principle of the IOU Tracker: with high accuracy detections at high frame rates, tracking can be done by simply associating detections by their spatial overlap between time steps.

With recent advances in the detection domain including CNN-based [4, 10, 5, 16] and traditional approaches with hand-crafted feature vectors [7, 9], new possibilities arise for tracking methods. Compared to previous approaches, gaps in the temporal stream of detections for an object are increasingly rare and the precision of the reported bounding boxes becomes very accurate. In combination with commonly higher frame rates of the video footage, e.g. 25 frames per second (fps) for the DETRAC dataset [17], also the differences in size and location of the detections have become significant smaller between frames.

All of these advances lead to a great simplification for the tracking task. It is therefore that in this paper a very simple tracking approach shall be assessed which is based on the idea of a passive detection filter introduced in [8]. Thanks to the mentioned performance improvements for detectors, we show that much simpler tracking approaches can be successful and the overhead from more sophisticated tracking algorithms is not necessarily needed in all cases.

Thanks to its very low computational footprint, the proposed method can serve as a simple baseline method for other trackers and allows an assessment of the importance of further efforts in the tracking algorithm. It further enables assessing tracking benchmarks in order to identify if the specific challenges they pose (e.g. missed detections, frame rate etc.) are in line with what algorithms already can achieve. The source code of the tracker is made publicly available[1].

2. Method

As mentioned above, both high precision detections and the usage of video footage with high frame rates can greatly simplify the tracking task. Our method is based on the assumption that the detector produces a detection per frame for every object to be tracked, i.e. there are none or only few "gaps" in the detections. Furthermore, we assume that detections of an object in consecutive frames have an unmistakably high overlap IOU (intersection-over-union) which is commonly the case when using sufficiently high frame rates. The IOU measure used in our approach is defined as

$$IOU(a,b) = \frac{Area(a) \bigcap Area(b)}{Area(a) \bigcup Area(b)}. \quad (1)$$

If both requirements are met, tracking becomes trivial and can be done even without using image information. We propose a simple IOU tracker which essentially continues a track by associating the detection with the highest IOU (see eq. 1) to the last detection in the previous frame if a certain threshold σ_{IOU} is met. All detections not assigned to an existing track will start a new one. All tracks without an assigned detection will end. This principle is also illustrated in Figure 1.

The performance is further improved by filtering out all tracks with a length shorter than t_{min} and the ones without at least one detection with a score above σ_h. Short tracks are removed because they usually root in false positives and generally add clutter to the output. Requiring a track to have at least one high-scoring detection ensures that the track belongs to a true object of interest while benefiting from low-scoring detections for the completeness of the track.

A detailed description of the method is shown in Algorithm 1 where D_f denotes the detections at frame f, d_j the j^{th} detection at that frame, T_a active tracks, T_f finished tracks and F the number of Frames in the sequence.

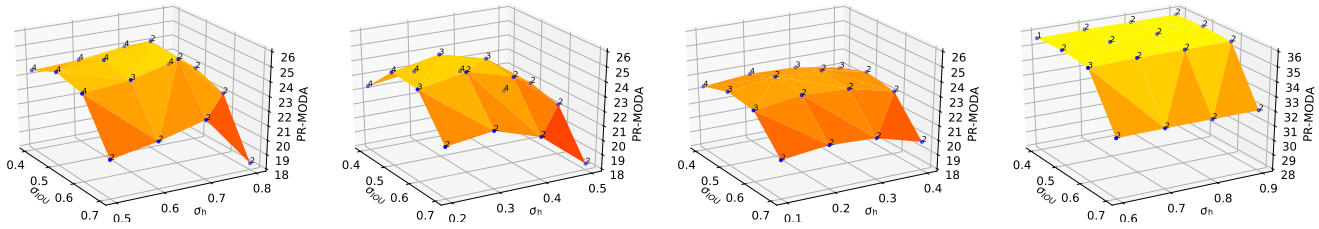
[1] https://github.com/bochinski/iou-tracker.git

Algorithm 1 IOU Tracker

1: **Inputs:**
   $D = \{D_0, D_1, ..., D_{F-1}\} =$
   $\{\{d_0, d_1, ..., d_{N-1}\}, \{d_0, d_1, ..., d_{N-1}\}, ...\}$
2: **Initialize:**
   $T_a = \emptyset, T_f = \emptyset$
   $D = \{\{d_i | d_i \in D_j, d_i \geq \sigma_l\} | D_j \in D\}$
3: **for** f = 0 to F :
4: **for** $t_i \in T_a$:
5: $d_{best} = d_j$ where $max(IOU(d_j, t_i)), d_j \in D_f$
6: **if** $IOU(d_{best}, t_i) \geq \sigma_{IOU}$:
7: add d_{best} to t_i
8: remove d_{best} from D_f
9: **else**
10: **if** $highest\_score(t_i) \geq \sigma_h$ and $len(t_i) \geq t_{min}$:
11: add t_i to T_f
12: remove t_i from T_a
13: **for** $d_j \in D_t$:
14: start new track t with d_j and insert into T_a
15: **for** $t_j \in T_A$:
16: **if** $highest\_score(t_i) \geq \sigma_h$ and $len(t_i) \geq t_{min}$:
17: add t_i to T_f
18: **return** T_f

Note that in line 5 only the best-matching, unassigned detection is taken as a candidate to extend the track. This does not necessarily lead to an optimal association between the detections D_f and tracks T_a but could be solved e.g. by applying the Hungarian algorithm maximizing the sum of all IOUs at that frame. However, taking the best match is a reasonable heuristic since σ_{IOU} is normally chosen in the same range as the IOU threshold for the non-maxima suppression of the detector. Therefore, multiple matches satisfying σ_{IOU} are rare in practice.

The overall complexity of the method is very low compared to other state-of-the-art trackers. No visual information of the frames is used, hence it can be seen as a simple filtering procedure on detection level. This means if the tracker is used on-line in conjunction with a state-of-the-art detector, the computational cost compared to the detectors becomes negligible. Therefore, tracks can be obtained at virtually no additional computational cost from the detections. If the tracker is performed standalone, frame rates exceeding 100K fps can be easily achieved as shown in the following experiments. It is also important to note that thanks to its speed, further tracking components can be added on top of its results e.g. by considering the output as tracklets which can be connected using image or motion information.



(a) R-CNN (b) ACF (c) CompACT (c) EB

Figure 2. Comparison of the PR-MOTA performance for different detectors and parameters. Each blue dot represents a measurement, the number shows the best value for t_{min}.

Detector	σ_{IOU}	σ_h	t_{min}	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDs	PR-FM	PR-FP	PR-FN	Speed
R-CNN	0.5	0.7	2	25.29%	44.38%	19.55%	**18.56%**	4721.16	4952.06	10172.98	157457.51	87,286 fps
ACF	0.5	0.3	3	24.92%	**44.71%**	20.27%	19.39%	**1755.71**	**2097.18**	12530.53	161241.24	**208,000 fps**
CompACT	0.5	0.2	2	23.41%	42.88%	18.39%	18.91%	2162.79	2284.12	7880.24	**152077.43**	204,140 fps
EB	0.5	0.8	2	**35.78%**	40.81%	**32.32%**	20.95%	4103.05	4195.60	**7745.71**	163944.54	12,880 fps

Table 1. Best settings and results for four state-of-the-art object detectors for the DETRAC-Train dataset.

3. Experiments

We examined the performance of the proposed tracker on the DETRAC dataset [17] consisting of over 10 hours of video footage targeting vehicle detection and tracking. They were recorded at 25 frames per second. Baseline detections for CompACT [5], R-CNN [10], ACF [7] and DPM [9] are available, although we do not report results based on the DPM detections since they are generally too inaccurate and therefore not suitable for our tracker. Furthermore, we computed additional detections using the Evolving Boxes detector (EB) [16] with the VGG16 1-3-5 model.

The evaluation is done using the UA-DETRAC evaluation protocol. For tracking, this means the method is run multiple times with different detection score thresholds σ_l to compute the precision-recall curve. Over this curve, the common CLEAR MOT metrics [14] are computed. The final scores are composed by the area under these curves and consider the performance of the tracker for all detector thresholds σ_l (see [17] fur further information). Note that this does not affect the thresholding with σ_h but rather the availability of low-scoring detections. In general and in accordance with [8], it can be assumed that a higher number of low-scoring detections would contribute to a higher tracking performance for our approach.

The implementation was done in pure Python without any performance optimizations.

The best parameters for σ_{IOU}, σ_h and t_{min} were determined by performing a grid search on the training dataset for each detector. The ranges of the search are shown in Table 2. Note that all detection scores were normalized to a range of $\sigma \in [0.0; 1.0]$ but are still differently distributed for each detector. Therefore, different ranges for σ_h have to be chosen.

All combinations of the three parameters within these ranges were evaluated, resulting in 64 runs per detector.

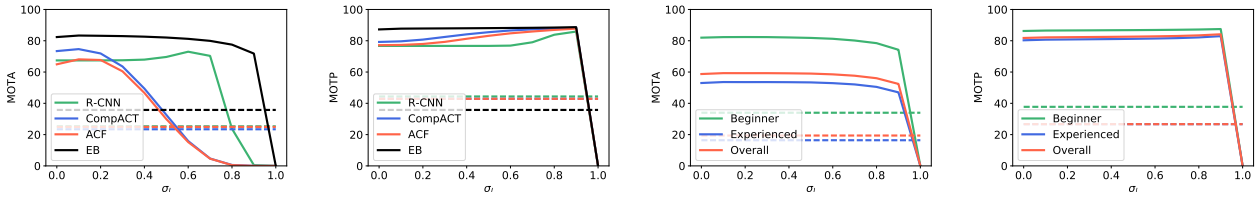
Detector	σ_{IOU}	σ_h	t_{min}
R-CNN	0.4 - 0.7	0.5 - 0.8	1 - 4
ACF	0.4 - 0.7	0.2 - 0.5	1 - 4
CompACT	0.4 - 0.7	0.1 - 0.4	1 - 4
EB	0.4 - 0.7	0.5 - 0.9	1 - 4

Table 2. Ranges for the grid-based parameter search for each detector. A step size of 1 is used for σ_{IOU} and 0.1 for σ_l and t_{min}.

The best configuration is chosen by the PR-MOTA metric as it is the primary metric in the UA-DETRAC challenge. A visualization of the results is shown in Figure 2, the best results for each detector and their respective configurations are compared in Table 3.

By far the best results are achieved using the EB detector with many near maximum scoring detections. It appears that these results also benefit from a potential flaw in the evaluation metric because the EB detector produces also a high amount of false positives with very low scoring detections. This effectively extends the PR curve to a low precision at a high recall. Our IOU tracker, however, is not affected by these detections but the area under the MOTA-over-PR curve becomes significantly larger. Consequently, a fair comparison would only be possible if the PR curve is fully defined between the intersections with the precision and recall axes.

Although CompACT shows a much better average precision of the PR curve (see [17] for further details) than the other reference detections, better PR-MOTA values can be achieved with ACF and R-CNN. This is because the CompACT detections are generally fewer but more precise than the ones from R-CNN and ACF. Our tracker however benefits from more detections since it is not able to predict missing detections. Especially in the DETRAC evaluation scripts, the detections are thresholded using σ_l before running the tracker. In case of no matching detection, this thus



(a) MOTA-Train (b) MOTP-Train (c) MOTA-Test (EB) (d) MOTP-Test (EB)

Figure 3. MOTA and MOTP Scores for different σ_l thresholds. The dotted lines represent the PR-MOTA/PR-MOTP scores for the respective method. The MOTA-Test and MOTP-Test values are generated using the EB detections.

Tracker	Detector	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDs	PR-FM	PR-FP	PR-FN	Speed
Overall (Easy + Medium + Hard)										
CEM [1]	CompACT	5.1%	35.2%	3.0%	35.3%	**267.9**	**352.3**	**12341.2**	260390.4	4.62 fps
CMOT [3]	CompACT	12.6%	36.1%	16.1%	18.6%	285.3	1516.8	57885.9	**167110.8**	3.79 fps
GOG [12]	CompACT	14.2%	37.0%	13.9%	19.9%	3334.6	3172.4	32092.9	180183.8	390 fps
DCT [2]	R-CNN	11.7%	38.0%	10.1%	22.8%	758.7	742.9	336561.2	210855.6	0.71 fps
H^2T [18]	CompACT	12.4%	35.7%	14.8%	19.4%	852.2	1117.2	51765.7	173899.8	3.02 fps
IHTLS [6]	CompACT	11.1%	36.8%	13.8%	19.9%	953.6	3556.9	53922.3	180422.3	19.79 fps
IOU	R-CNN	16.0%	**38.3%**	13.8%	20.7%	5029.4	5795.7	22535.1	193041.9	**100,840 fps**
IOU	EB	**19.4%**	28.9%	**17.7%**	**18.4%**	2311.3	2445.9	14796.5	171806.8	6,902 fps
Easy (AVSS17 Beginner Challenge)										
CEM [1]	Average	7.2%	42.0%	3.4%	42.3%	96.4	119.6	4253.3	63296.3	-
CMOT [3]	Average	16.6%	43.8%	20.0%	23.0%	**68.3**	327.9	16282.2	39467.6	-
GOG [12]	Average	20.1%	44.6%	17.5%	24.0%	1019.0	981.2	8423.5	42065.9	-
DCT [2]	Average	16.3%	44.1%	9.6%	34.3%	73.5	**69.4**	4754.6	51393.3	-
H^2T [18]	Average	17.1%	42.9%	17.8%	24.7%	298.8	305.5	12866.0	42086.5	-
IHTLS [6]	Average	14.8%	43.0%	15.5%	25.3%	299.5	1102.3	13839.9	43954.4	-
IOU	R-CNN	29.3%	**47.2%**	25.0%	**17.3%**	1112.5	1261.0	3457.6	**33394.1**	**117,340 fps**
IOU	EB	**34.0%**	37.8%	**27.9%**	20.4%	573.6	603.7	**1617.0**	33760.8	9,002 fps
Medium + Hard (AVSS17 Experienced Challenge)										
IOU	R-CNN	11.8%	**36.5%**	8.9%	25.0%	3693.1	4228.3	16634.7	168527.2	**87,906 fps**
IOU	EB	**16.4%**	26.7%	**14.8%**	**18.2%**	**1743.2**	**1846.3**	**12627.0**	**136077.8**	6,069 fps

Table 3. Best results for each Tracker on the DETRAC-Test dataset for overall, easy and medium+hard sets. For the easy split, only average scores over all four detectors are available. The medium and hard sets can only be evaluated together in conjunction with the AVSS17 challenge, but no baseline results are provided. Except for the IOU tracker the results were taken from [17].

inhibits searching for detections with $\sigma < \sigma_l$ which could be a significant improvement according to [8].

Therefore, a single missed detection for one track results both in an ID switch and a false negative, lowering the overall performance. On the other hand, false positives can be ruled out to some extent since they usually produce short tracks consisting of low-scoring detections. Such tracks will be filtered out using the thresholds for the high scoring detection with σ_h and minimum length t_{min}.

Based on this evaluation, our tracker is tested with the R-CNN detections with $\sigma_{IOU} = 0.5$, $\sigma_h = 0.7$ and $t_{min} = 2$ and EB detections with $\sigma_{IOU} = 0.5$, $\sigma_h = 0.8$ and $t_{min} = 2$ on the DETRAC-Test data. A comparison of the results to six state-of-the-art trackers is shown in Table 3.

Our IOU tracker outperforms the other methods considerably with respect to the overall metrics for accuracy (PR-MOTA) and precision (PR-MOTP) as well as mostly tracked (PR-MT) and mostly lost (PR-ML). Furthermore, a speed of over 100K fps is achieved in case of the R-CNN detections, which is magnitudes faster than the baseline methods with 0.7-390 fps. The high amount of high scoring detections for EB greatly increases the number of detections to process on a larger range of varying σ_l, thus decreasing the achievable number of fps. Nonetheless, even with EB detections, the runtime is negligible compared to the other trackers.

The huge difference of the performance between the training and the overall test data is probably related to the different accuracy of the detectors between those sets. Since the detectors were also trained on the training sequences it is obvious that they produce better results on these videos than on the test sequences. As our tracker is prone to detection errors, the performance drops as well. Additionally, the parameters trained on the training data may suffer from overfitting to the high quality detections on the training data.

Figure 3 shows the MOTA and MOTP scores for the different values of σ_l which are summarized in the final PR-MOTA and PR-MOTP scores. The plots show that the tracker can handle a wide range of σ_l scores while the MOTA value changes only slightly. The performance starts dropping only after rising above σ_h but not in the low σ_l range which is thanks to the filtering of tracks without detections of at least a score of σ_h. MOTP shows a tendency of rising with σ_l as for this metric, high values are obtained with more accurate detections.

The evaluation of the easy sequences in Table 3 shows

Tracker	MOTA	MOTP	FP	FN
$Best$ [15]	71.0	80.2	7,880	44,564
IOU_{SDP}	57.1	77.1	5,702	70,278
IOU_{FR-CNN}	45.4	77.5	7,639	89,535
$Average$	44.3	76.4	8,372	92,128

Table 4. Comparison of the results of the IOU tracker based on SDP and FR-CNN detections to the best performing method according to the MOTA score and the overall average of the MOT16 test dataset.

superior results over the state-of-the art but care must be taken when interpreting these numbers for the reference methods as they represent only the average scores over all four detectors, which is not imitable for us since the UA-DETRAC evaluation server submission policy prohibits excessive testing on the test data.

The experiments show that in some cases such as the DETRAC dataset, simple tracking methods like the IOU tracker can lead to better results than complex approaches based on decades of research. However, this is not universally valid but depends on the dataset used. In other tracking tasks, like pedestrian tracking, the size and aspect ratios usually undergo greater changes in only few frames, e.g. because a person is walking. Heavier occlusions and lower frame rates can also reduce the success rate for correctly matching detections by calculating their overlap, indicating the requirement for more sophisticated methods.

With these considerations in mind, we evaluated the performance of the IOU tracker on the MOT16 / MOT17 benchmark [11] for pedestrian tracking using the provided Faster FR-CNN [13] and SDP [19] detections. The frame rates of the sequences range from 14 to 30 fps. The best parameters for the method are determined by an extensive grid search over the training dataset. Since there are only 7 train sequences, a complete sweep over the parameter space in 0.1 steps for $\sigma_h, \sigma_l, \sigma_{IOU}$ and $t_{min} \in \{1, 2, 3, 4, 5\}$ was feasible. The best parameters using the FR-CNN detections are $\sigma_h = 0.9$, $\sigma_l = 0.0$, $\sigma_{IOU} = 0.4$, $t_{min} = 4$ achieving a MOTA score of 49.96. For SDP, $\sigma_h = 0.5$, $\sigma_l = 0.3$, $\sigma_{IOU} = 0.3$, $t_{min} = 5$ was best with a MOTA score of 62.77. The results on the test sequences are shown in Table 4. It can be seen that the IOU tracker with FR-CNN detections already achieves a performance slightly above the average. With the more accurate SDP detections, especially the MOTA score can be boosted considerably and places the tracker on place 13 out of 64 at the time of writing this paper. This shows that even with the more challenging conditions of pedestrian tracking, moving cameras and various frame rates, competitive results can be achieved.

Additionally, our experiments show that in the case of vehicle tracking and dealing with fixed-sized objects, static cameras and high accuracy detections at high frame rates, good tracking can be achieved on a simple level. We recommend that the results of this tracking approach be taken into consideration for the design of new tracking benchmarks.

4. Conclusions

In this paper, we showed that with simple means successful tracking can be done. Our presented IOU tracker considerably outperforms the state-of-the-art at only a fraction of the complexity and computational cost. This becomes possible due to the recent advances in the object detection domain, not at least due to the current boom of CNN-based approaches. In combination with commonly higher frame rates of videos, the requirements for a multi-object tracker in a tracking-by-detection framework drastically changed. Our simple, yet effective IOU tracker exploits these traits and could serve as an example to reflect the design of a tracker within those new conditions.

Acknowledgements

The research leading to these results has received funding from the European Community's FP7 and BMBF-VIP+ under grant agreement number 607480 (LASIE) and 03VP01940 (SiGroViD).

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1272. IEEE, 2011.

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1926–1933. IEEE, 2012.

[3] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1218–1225, 2014.

[4] E. Bochinski, V. Eiselein, and T. Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 278–285, Colorado Springs, CO, USA, Aug. 2016.

[5] Z. Cai, M. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3361–3369, 2015.

[6] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2304–2311, 2013.

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[8] V. Eiselein, E. Bochinski, and T. Sikora. Assessing post-detection filters for a generic pedestrian detector in a tracking-by-detection scheme. In *Analysis of video and audio "in the Wild" workshop at IEEE AVSS17*, Lecce, Italy, Aug. 2017.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

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

[12] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208. IEEE, 2011.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[14] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan. The clear 2006 evaluation. *Multimodal Technologies for Perception of Humans*, pages 1–44, 2007.

[15] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multi people tracking with lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue. Evolving boxes for fast vehicle detection. *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2017.

[17] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015.

[18] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1282–1289, 2014.

[19] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, 2016.