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 and additional implementation and training details of its object detector and tracking 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

For the sake of completeness and in order to facilitate the reproduction of our results, we provide additional implementation details and references of our Tracktor and its extensions.

A.1. Tracktor

In Algorithm 1 and 2, we present a structured pseudocode representation of our Tracktor 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, 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) [18] multi-object detector with Feature Pyramid Networks (FPN) [16] on the MOT17Det [17] dataset.

In addition, we follow the improvements suggested by [2]. These include a replacement of the Region of Interest (RoI) pooling [7] by the *crop and resize* pooling suggested by Huang et al. [10] and training with a batch size of $N = 1$ instead of $N = 2$ while increasing the number of extracted regions from $R = 128$ to $R = 256$. These changes and the addition of FPN ought to improve the detection results for comparatively small objects. We achieve the best results with a ResNet-101 [8] as the underlying feature extractor. In Table 1, we compare the performance 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 [6], FRCNN and SDP [20] (ascending order). Our FRCNN implementation without FPN is on par with the official MOT17Det entry and represents the detector applied in the *Tracktor-no-FPN* variant of our ablation study in Section 3.1.

Method	AP ↑	MODA ↑	FP ↓	FN ↓	Precision ↑	Recall ↑
FRCNN + FPN	0.81	70.2	14914	19196	**96.5**	83.2
FRCNN	0.72	71.6	8227	24269	91.6	78.8
DPM [6]	0.61	31.2	42308	36557	64.8	68.1
FRCNN [18]	0.72	68.5	10081	25963	89.8	77.3
SDP [20]	**0.81**	**76.9**	**7599**	**18865**	92.6	**83.5**

Table 1: A comparison of our Faster R-CNN (FRCNN) with Feature Pyramid Networks (FPN) implementation on the MOT17Det detection benchmark with the three object detection methods mentioned in this work. Our vanilla FRCNN results are on par with the official FRCNN implementation. The extension with FPN yields a detection performance close to SDP. 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 [5]. 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

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

unsteady camera movement. In the case of an additional camera translation such as in the autonomous driving sequences of 2D MOT 2015 [17], we resort to the affine transformation. It should be noted that in MOT17 [17], 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 $t-1$. The CVA motion model is only applied to the *AVG-TownCentre* sequence of 2D MOT 2015.

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 [9] architecture which is based on ResNet-50 [8] with the triplet loss and *batch hard* strategy as presented in [9]. The network is optimized with Adam [13] with $\beta = (0.9, 0.999)$ and a decaying learning rate as described in [9]. 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 3. 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) [11] and ID F1 Score (IDF1) [19]. However, previous Tables such as 3 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 [17] 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 [12, 1, 3] and their own classification score and NMS procedure. Therefore, we consider the comparison with public trackers as fair.

B.3. Evaluation on public detections

By reclassifying and regressing the given public detections with a private object detector, Tracktor reduces the equalizing effect of public detections to the initialization of new tracks. In addition to our remarks in Section 3 regarding the *publicness* of our method, we emphasize the potential of Tracktor in comparison with other state-of-the-art trackers even without the advantage of the reclassification and regression. To this end, we show Table 2, which evaluates all trackers on the MOT17 test set only with Faster R-CNN public detections. Tracktor-no-FPN++ (without Feature Pyramid Networks) uses a vanilla Faster R-CNN for reclassification and regression, effectively, not altering the public detections. However, the results support the overall conclusions from Table 2 of our main work.

B.4. 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

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
Tracktor++	42.14	45.76	18.17	38.93	3918	83904	648
Tracktor-no-FPN++	**39.41**	43.46	16.63	39.00	**6975**	**83380**	922
eHAF17	37.37	**46.44**	**20.63**	35.83	11050	86510	605
FWT	39.06	42.07	17.60	37.53	8397	88290	780
jCC	37.64	46.66	18.70	36.33	9984	86897	**577**
MOTDT17	38.81	46.34	14.47	36.91	8911	88773	731
MHT_DAM	37.54	46.17	17.43	**34.86**	9795	89294	742

Table 2: Comparison on MOT17 test set with Faster R-CNN public detections. Tracktor-no-FPN++ applies vanilla Faster R-CNN.

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.

B.5. Tracktor video frame rate robustness

A successful Tracktor bounding box regression depends on sufficiently high video frame rates or, in other words, small frame-by-frame object displacements. A possible approach to address this issue is the extension with a powerful motion model. A rudimentary motion model, the camera motion compensation (CMC), is presented in Section 2.3 and evaluated in the ablation study in Table 1. However, MOT16 and MOT17 mostly consist of sequences with benevolent video frame rates and slow moving objects (pedestrians).

We therefore complement our analysis of Tracktor in challenging tracking scenarios from Section 4.1 with an evaluation of its video frame rate robustness. To this end, we evaluate Tracktor and Tracktor++ on all MOT17 training sequences with originally 30 frames per second (FPS) and reduce their frame rates by removing frames from the data and ground truth. In Figure 1, both versions exhibit a fairly robust object tracking (MOTA) and identity preservation (IDF1) for rates as low as 5 FPS. As expected, the performance for very small rates suffers particularly with respect to identity preservation.

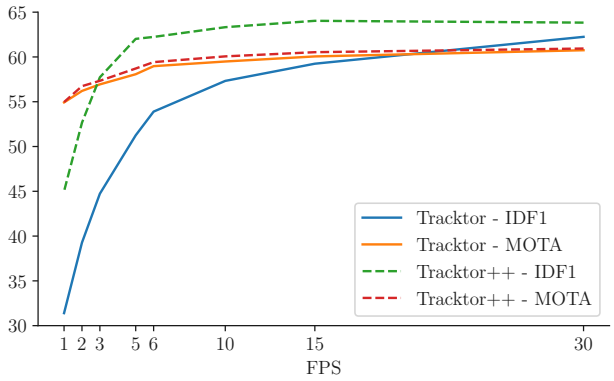

Figure 1: Tracking performance of Tracktor and Tracktor++ on low frame rate versions of the MOT17-{02, 04, 09, 10, 11}-FRCNN sequences.

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 vanilla Tracktor 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 [14] algorithm. In the case of an object-object occlusion (IoU > 0.8), the ground truth matching is applied 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 regression oracle then sets track bounding boxes to the corresponding ground truth coordinates at frame t.

- **Oracle-MM:** A perfect motion model works analogous to Oracle-REG but we only 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:** Again, we use the Hungarian algorithm to match the new set of detections to the ground truth data. Ground truth identity matches between inactive tracks and new detections yield a perfect re-identification.

Algorithm 1: Tracktor algorithm (private detections)

Data: Video sequence as ordered list $I = \{i_0, i_1, \cdots, i_{T-1}\}$ of images i_t.

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

$\mathcal{T}, \mathcal{T}_{\text{active}} \leftarrow \emptyset$;
for $i_t \in I$ **do**
  $B, S \leftarrow \emptyset$;
  for $T_k \in \mathcal{T}_{\text{active}}$ **do**
    $\mathbf{b}_{t-1}^k \leftarrow T_k[-1]$;
    $\mathbf{b}_t^k, s_t^k \leftarrow \text{detector.reg\_and\_class}(\mathbf{b}_{t-1}^k)$;
    if $\mathrm{s}_t^k < \sigma_{\text{active}}$ **then**
      $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\}$;
      $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\}$;
    else
      $B \leftarrow B + \{\mathbf{b}_t^k\}$;
      $S \leftarrow S + \{\mathrm{s}_t^k\}$;
  $B \leftarrow \texttt{NMS}(B, S, \lambda_{\text{active}})$;
  for $k, T_k \in \mathcal{T}_{\text{active}}$ **do**
    if $k \notin B$ **then**
      $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\}$;
      $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\}$;
  for $T_k, \mathbf{b}_t^k \in \texttt{zip}(\mathcal{T}_{\text{active}}, B)$ **do**
    $T_k \leftarrow T_k + \{\mathbf{b}_t^k\}$;
  $\mathcal{D}_t \leftarrow \text{detector.detections}(i_t)$;
  for $d_t \in \mathcal{D}_t$ **do**
    for $\mathbf{b}_t^k \in B$ **do**
      if $\texttt{IoU}(d_t, \mathbf{b}_t^k) > \lambda_{\text{new}}$ **then**
        $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{d_t\}$;
  for $d_t \in \mathcal{D}_t$ **do**
    $T_k \leftarrow \emptyset$;
    $T_k \leftarrow T_k + \{d_t\}$;
    $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} + \{T_k\}$;
$\mathcal{T} \leftarrow \mathcal{T} + \mathcal{T}_{\text{active}}$;

Algorithm 2: Tracktor algorithm (public detections)

Data: Video sequence as ordered list $I = \{i_0, i_1, \cdots, i_{T-1}\}$ of images i_t and public detections as ordered list $\mathcal{D} = \{\mathcal{D}_0, \mathcal{D}_1, \cdots, \mathcal{D}_{T-1}\}$ of detections $\mathcal{D}_t$.

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

$\mathcal{T}, \mathcal{T}_{\text{active}} \leftarrow \emptyset$;
for $i_t, \mathcal{D}_t \in \texttt{zip}(I, \mathcal{D})$ **do**
  $B, S \leftarrow \emptyset$;
  for $T_k \in \mathcal{T}_{\text{active}}$ **do**
    $\mathbf{b}_{t-1}^k \leftarrow T_k[-1]$;
    $\mathbf{b}_t^k, s_t^k \leftarrow \text{detector.reg\_and\_class}(\mathbf{b}_{t-1}^k)$;
    if $\mathrm{s}_t^k < \sigma_{\text{active}}$ **then**
      $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\}$;
      $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\}$;
    else
      $B \leftarrow B + \{\mathbf{b}_t^k\}$;
      $S \leftarrow S + \{\mathrm{s}_t^k\}$;
  $B \leftarrow \texttt{NMS}(B, S, \lambda_{\text{active}})$;
  for $k, T_k \in \mathcal{T}_{\text{active}}$ **do**
    if $k \notin B$ **then**
      $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} - \{T_k\}$;
      $\mathcal{T} \leftarrow \mathcal{T} + \{T_k\}$;
  for $T_k, \mathbf{b}_t^k \in \texttt{zip}(\mathcal{T}_{\text{active}}, B)$ **do**
    $T_k \leftarrow T_k + \{\mathbf{b}_t^k\}$;
  $S \leftarrow \emptyset$;
  for $\mathbf{d}_t \in \mathcal{D}_t$ **do**
    $\mathbf{d}_t, s_t \leftarrow \text{detector.reg\_and\_class}(\mathbf{d}_t)$;
    if $s_t < \sigma_{\text{active}}$ **then**
      $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{\mathbf{d}_t\}$;
    else
      $S \leftarrow S + \{s_t\}$;
  $\mathcal{D}_t \leftarrow \texttt{NMS}(\mathcal{D}_t, S, \lambda_{\text{new}})$;
  for $\mathbf{d}_t \in \mathcal{D}_t$ **do**
    for $\mathbf{b}_t^k \in B$ **do**
      if $\texttt{IoU}(d_t, \mathbf{b}_t^k) > \lambda_{\text{new}}$ **then**
        $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{d_t\}$;
  for $d_t \in \mathcal{D}_t$ **do**
    $T_k \leftarrow \emptyset$;
    $T_k \leftarrow T_k + \{d_t\}$;
    $\mathcal{T}_{\text{active}} \leftarrow \mathcal{T}_{\text{active}} + \{T_k\}$;
$\mathcal{T} \leftarrow \mathcal{T} + \mathcal{T}_{\text{active}}$;

Sequence	Detection	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
			MOT17 [17]					
MOT17-01	DPM [6]	35.9	37.1	20.8	50.0	131	3962	39
MOT17-03	DPM	65.2	57.0	35.1	12.8	1338	34840	222
MOT17-06	DPM	52.7	55.7	18.5	40.1	184	5310	80
MOT17-07	DPM	40.5	42.5	10.0	40.0	363	9603	90
MOT17-08	DPM	27.0	30.7	9.2	50.0	213	15130	83
MOT17-12	DPM	45.6	55.2	16.5	48.4	88	4596	29
MOT17-14	DPM	26.9	37.1	6.7	53.0	591	12834	92
MOT17-01	FRCNN [18]	34.9	34.8	20.8	41.7	406	3753	39
MOT17-03	FRCNN	66.4	59.7	37.2	13.5	1014	33961	189
MOT17-06	FRCNN	56.7	59.0	23.0	27.5	359	4647	96
MOT17-07	FRCNN	39.4	43.1	11.7	40.0	555	9588	93
MOT17-08	FRCNN	27.1	31.7	11.8	50.0	197	15119	74
MOT17-12	FRCNN	43.4	53.9	15.4	51.6	185	4697	25
MOT17-14	FRCNN	27.1	38.1	7.3	48.2	1202	12139	132
MOT17-01	SDP [20]	37.5	36.8	25.0	41.7	283	3706	42
MOT17-03	SDP	69.6	60.1	39.9	10.8	2469	29065	248
MOT17-06	SDP	56.8	59.2	26.1	28.8	354	4638	93
MOT17-07	SDP	41.2	42.6	11.7	33.3	596	9231	111
MOT17-08	SDP	28.7	32.1	13.2	47.4	253	14715	103
MOT17-12	SDP	45.3	56.9	18.7	48.4	212	4492	34
MOT17-14	SDP	27.6	38.5	7.3	48.2	1208	12021	158
All		53.5	52.3	19.5	36.6	12201	248047	2072
			MOT16 [17]					
MOT16-03	DPM	65.8	57.9	35.1	12.2	1397	34101	226
MOT16-06	DPM	53.9	57.9	20.4	39.4	243	5000	80
MOT16-07	DPM	43.0	43.6	13.0	33.3	405	8808	97
MOT16-08	DPM	34.3	36.8	12.7	38.1	314	10577	101
MOT16-12	DPM	48.0	57.0	18.6	44.2	108	4172	30
MOT16-14	DPM	27.4	37.6	6.7	51.2	659	12645	108
All		54.4	52.5	19.0	36.9	3280	79149	682
			2D MOT 2015 [15]					
TUD-Crossing	ACF [4]	78.3	58.3	53.8	0.0	14	207	18
PETS09-S2L2	ACF	44.5	28.4	4.8	2.4	644	4420	289
ETH-Jelmoli	ACF	57.8	67.4	35.6	24.4	317	732	21
ETH-Linthescher	ACF	49.3	55.5	15.7	50.8	178	4303	48
ETH-Crossing	ACF	43.0	54.2	11.5	38.5	22	538	12
AVG-TownCentre	ACF	39.0	38.5	17.3	19.0	620	3075	665
ADL-Rundle	ACF-1	33.7	49.3	28.1	9.4	2497	3615	56
ADL-Rundle	ACF-3	45.6	46.0	15.9	13.6	750	4713	68
KITTI-16	ACF	48.1	50.8	17.6	5.9	174	672	37
KITTI-19	ACF	49.4	59.5	14.5	14.5	553	2082	71
Venice-1	ACF	35.1	42.6	23.5	29.4	708	2220	33
All		44.1	46.7	18.0	26.2	6477	26577	1318

Table 3: 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.

References

[1] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification, 07 2018. 2

[2] Xinlei Chen and Abhinav Gupta. An implementation of faster RCNN with study for region sampling. *CoRR*, abs/1702.02138, 2017. 1

[3] Young chul Yoon, Abhijeet Boragule, Young min Song, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *AVSS*, 2018. 2

[4] Piotr Dollar, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *PAMI*, 36(8):1532–1545, Aug. 2014. 5

[5] Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *PAMI*, 30(10):1858–1865, 2008. 1

[6] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *pami*, 32:1627–1645, 2009. 1, 5

[7] Ross B. Girshick. Fast r-cnn. *ICCV*, pages 1440–1448, 2015. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, abs/1512.03385, 2015. 1, 2

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 2

[10] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CVPR*, abs/1611.10012, 2016. 1

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

[12] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *PAMI*, pages 1–1, 2018. 2

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2

[14] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955. 3

[15] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942*, 2015. 5

[16] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[17] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 1, 2, 5

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NIPS)*, 2015. 1, 5

[19] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. *ECCV Workshops*, 2016. 2

[20] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *CVPR*, pages 2129–2137, 2016. 1, 5