A Locality-Aware City-Scale Multi-Camera Vehicle Tracking System

Yunzhong Hou Heming Du Liang Zheng
Australian National University

Abstract

Vehicle tracking across multiple cameras can be difficult for modern tracking systems. Given unlikely candidates and faulty similarity estimation, data association struggles at city-scale tracking. In order to avoid difficulties in a large scenario, we keep the tracking procedure within a minimal range. The benefit of this smaller scenario idea is two-fold. On the one hand, ruling out most unlikely candidates decrease the possibility of mis-assignment. On the other hand, the system can devote all its discriminative power on the remaining local candidate pool. In fact, our tracking system features two parts to keep the data association within a small range, while at the same time increase the locality awareness for smaller scenarios. First, multiple cues including spatial-temporal information and camera topology are leveraged to restrict the candidate selection. Second, the appearance similarity estimation module is carefully tuned so that it focuses on the smaller local candidate pool. Based on a minimal view for the large scenario, the proposed system finished 5^{th} place in the 2019 AI-City challenge for city-scale multi-camera vehicle tracking.

1. Introduction

City-scale multi-camera vehicle tracking [19] is a subset of multi-target multi-camera tracking (MTMCT), which focuses on tracking multiple targets across different cameras [15]. However, given its vast coverage, huge candidate pool, and poor similarity estimation, city-scale vehicle tracking needs a dedicated system for best performance.

Many researchers focus on human MTMCT since the introduction of DukeMTMC dataset [15]. These tracking systems usually adopt a tracking-by-detection [5] workflow. However, human tracking systems are not ready for application on city-scale vehicle tracking directly. In fact, vehicle MTMCT differs from human counterpart in both candidate pool size and similarity estimation accuracy.

First, city-scale vehicle tracking systems cover a vast area. One scenario in the 2019 AI-City challenge [19] spans more than 2 kilometers, whereas DukeMTMC dataset only covers a university campus. Moreover, vehicle tracking system contains many distractors, including parked cars. Unlike humans who rarely stands at the same place for a prolonged period of time, these parked cars seldom move. In the evaluation protocol, these parked cars are not included either. Second, vehicles share similar appearance which is fairly difficult to re-identify even for a human. Vehicles only have a handful of types and the most common types share a lot of visual similarities. To make matter worse, the appearance of the same vehicle in different perspectives diverse significantly.

In the proposed system, we tackle the problem through minimal candidate selection and locality-aware similarity estimation. On one hand, we rule out the unlikely candidates. We start by refining the region-of-interest (ROI) in detection selection to rule out the parked cars. Then, when associating tracklets into single-camera trajectories, smoothness scores are used to rule out unlikely tracklet pairs. After the completion of single camera trajectories, the cross camera data association is first conducted on smaller sub-scenarios covering a single intersection, then city-wide. On the other hand, we increase the appearance discriminative power for smaller scenarios. The re-ID feature extractor is still trained on a global dataset, but the similarity estimation module in the vehicle MTMCT system is fine-tuned. Since the appearance of vehicles is most similar within cameras, we set a more acute threshold for single camera tracking, and a more robust threshold for cross camera tracking. At last, data association is conducted within a smaller range and a locality-aware similarity score.

We make the following contributions to city-scale vehicle MTMCT in the proposed system.

- Candidate selection with various cues including ROI refinement, spatial-temporal smoothness, and sub-scene division.
- A locality-aware similarity estimation module, with more acute parameter setting for the reduced candidate pool.
- A high-performing system on the AI-City 2019 challenge city-scale vehicle MTMCT based on merely the provided detection and training data.

Figure 1: Output of the proposed vehicle MTMCT system. The green cab is successfully tracked across multiple cameras.

2. Related Work

Multi-object tracking (MOT) systems usually follow the tracking-by-detection idea [5]. With the introduction of the renowned MOT challenge datasets [6, 13], this area draws attention from many researchers. Currently, most state-of-the-art MOT systems [12, 21, 10] features a detection proposal and selection part. In fact, the proposed DPM detection [2] provides noisy results. As for remedy, these top-performing MOT trackers generate detection proposals by themselves. For similarity estimation, MOT systems usually use Euclidean distance between person re-identification (re-ID) [25] features. As for data association, either the online methods or the offline methods are adopted. For online methods, only information form current or past time slots can be used. These online methods usually greedily associate detection in the current time slot into tracklets [10]. For offline methods, batch optimization methods are usually adopted since they can benefit from the information in future time slots. For example, in [12, 21], researchers choose correlation clustering, and in [18], researchers formulate the problem as lifted multicut.

Cross-camera tracking is also a vital part of MTMCT. Most researchers associate cross-camera identities based on single-camera trajectories [15, 16, 24]. Person re-ID methods also adopted in top-performing trackers for best performance [16, 24]. As for data association, in [23], the authors choose the online method with track-hypothesis trees. In [15, 16, 24], the author adopts the offline method through correlation clustering.

Vehicle tracking is an emerging aspect of MTMCT. Since the 2018 AI-City challenge [14], many researchers focus on this new problem. Most of the systems follow the tracking-by-detection method [20, 22, 3, 1]. Specifically, in [20], Tang *et al*. use multiple cues including trajectory smoothness, velocity change and temporal information for single-camera tracking. For cross-camera tracking, the researchers seek deep learning features, license plate features, and detected car type. All these efforts combined results in the top-performing result in last year AI-City challenge winner for tracking. In [22], researchers propose an adaptive-feature-learning technique for vehicle re-ID feature learning. In [3], Feng *et al*. use trajectories distance to aid the cross camera tracking procedure.

3. Method

3.1. System Overview

In the proposed system, short but reliable tracklets are first computed from detections. Next, these tracklets are linked into single-camera trajectories. Then, in each subscene covering a single corner or crossing, trajectories are

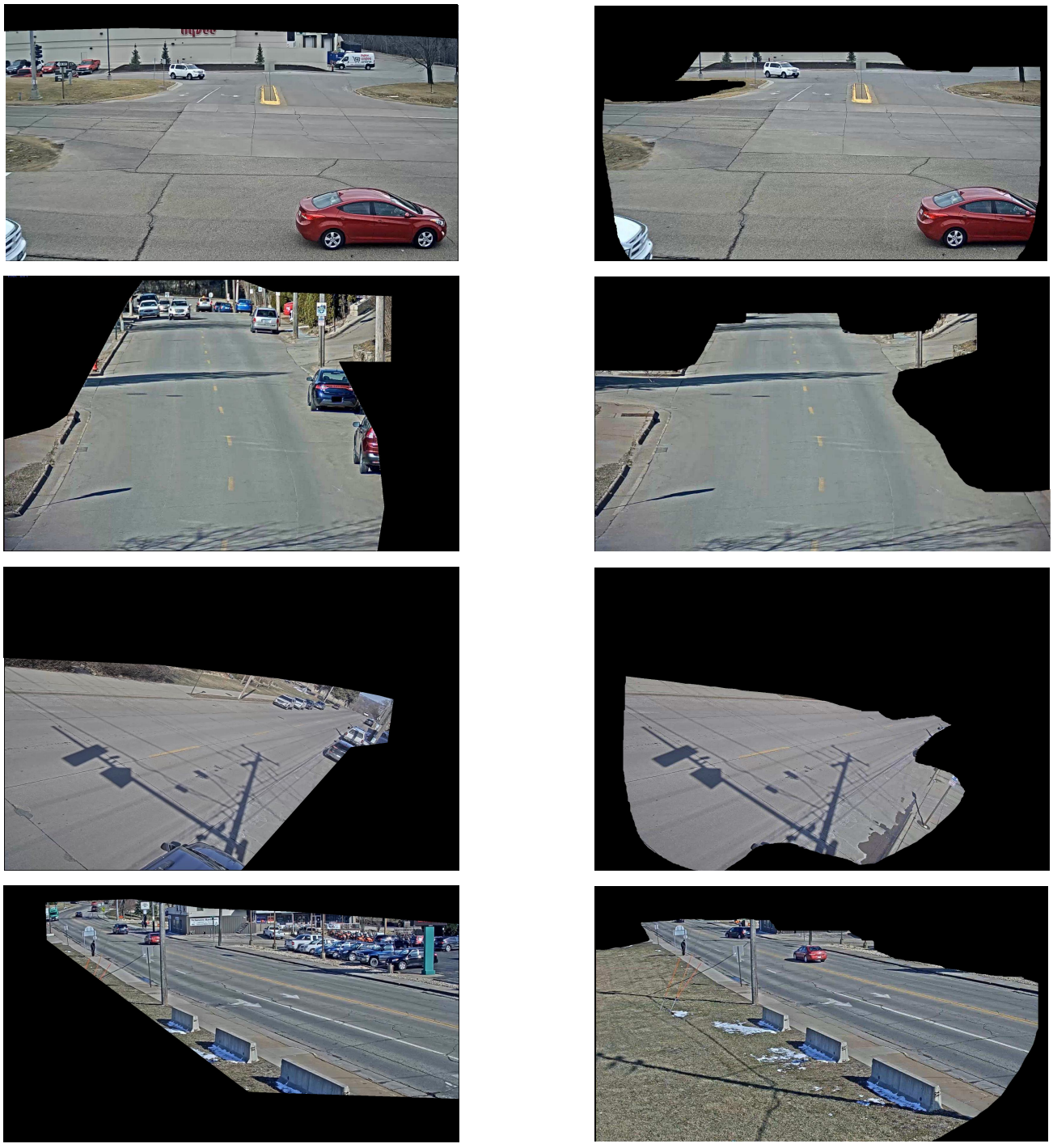
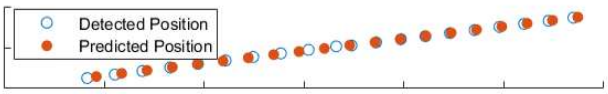


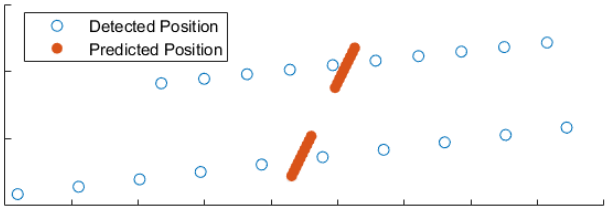
Figure 2: Our ROI refinement for detection selection.

merged into cross-camera identities. At last, the cross-camera identities in each sub-scene are associated together.

Problem formulation. We follow the same formulation as in [15]. In each of the four afore-mentioned association steps, candidates are formulated as nodes/vertices V, and the correlation weights from re-ID features are denoted as edges E. Given candidates and pair-wise correlation weights, a graph $G = (V, E)$ is constructed.



(a) Possible candidate pair based on smoothness.



(b) Impossible candidate pair based on smoothness.

Figure 3: Smoothness score for tracklets pair selection.

For a pair of nodes $i, j \in V$, $x_{i,j} \in \{-1, 1\}$ is a indicator of whether they are of the same identity. The optimization problem is formulated as follows,

$$\max_{x_{i,j}} \sum_{\forall i,j \in V, w_{i,j} \in E} x_{i,j} w_{i,j}. \tag{1}$$

Eq. 1 maximizes intra-group similarity, while at the same time minimizes inter-group similarity.

Person detection. We adopt the single shot multibox detector [8] provided in the AI-City development kit.

Similarity estimation. Given a pair of CNN features $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, their appearance similarity score is computed as,

$$w_{i,j} = \frac{thres - d\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right)}{norm}, \tag{2}$$

where $d(\cdot)$ is the Euclidean distance metric. $thres$ and $norm$ are hyper parameters we choose specifically to best match the smaller scenario.

Data association. We choose correlation clustering methods in our offline tracking system. Candidates are associated according to their pair-wise similarities as computed in Eq. 2.

3.2. Minimal Candidate Selection

The idea of minimal candidate selection is applied throughout the tracking procedure. First, we refine the ROI in detection selection. Next, when associating single camera trajectories, smoothness scores are used to rule out unlikely pairs. After that, the cross camera data association is first conducted on smaller sub-scenarios and then city-wide.

ROI refinement. As shown in Fig. 2, the ROI provided by AI-City challenge contains parked cars. These cars do not move for the whole video duration. Besides, the evaluation protocol does not contain these parked cars. Since these parked only increase candidate pool size and are not included in the evaluation protocol, they can be safely removed from the system without any loss. To do this, we carefully design the ROI in every camera to best exclude the parked cars as well as other false positive detections.

Figure 4: Overview of scenario S02. S02 has only one intersection. Thus, it only contains one sub-scene.

Smoothness score for trajectories association. As introduced in Section 3.1, candidates are associated according to their pair-wise similarities. When associating tracklets into single-camera trajectories, smoothness of the tracklets is leveraged to rule out the unlikely pairs. First, the car in the camera plane is projected into the map plane according to the provided homography matrix. After projection, the car bounding box position in the camera plane will be denoted by the longitude and latitude of the car. For better readability, we use the radius of the earth to further project the longitude and latitude of the car into an X-Y coordinate system. The origin point of that X-Y coordinate system is set to the center of each scenario. In the following parts, the car position refers to its position on the map. Then, for every pair of tracklets, the smoothness of their car position is calculated as the average difference between the detected position and predicted position. Polynomial curve fitting is employed to estimate the predicted position. If the average position difference is larger than 2 meters, this pair is marked as impossible.

Sub-scene division. In the 2019 AI-City challenge, two scenarios are used for testing. S02 (as shown in Fig. 4) is a highway intersection featuring four cameras. S05, on the other hand, contains 5 intersections and 19 cameras. Directly considering all the cameras in S05 within a 2-kilometer-spanning area can include many unlikely candidate pairs. To avoid this, we divide S05 into 6 sub-scenarios. Each of the 6 sub-scenarios contains cameras near an intersection, and there are at least 100-meter-long

Figure 5: Overview of the city-scale scenario S05. S05 spans more than 2 kilometers and is divided into multiple sub-scenario covering only one corner/crossing.

gaps in between these sub-scenarios. S02 is considered the only sub-scene in itself. When connecting the single-camera trajectories, we first consider the sub-scenarios. Then, the cross camera identities in each sub-scenarios are linked together.

3.3. Locality Aware Similarity Estimation

The similarity estimation module in the proposed only base on the re-ID features. With carefully design and tuning, it provides reliable performance for the refined candidate pool in both single camera tracking and cross camera tracking.

Vehicle re-ID features. We use only the provided re-ID dataset in AI-City (track-2) to train an appearance feature extractor. The base network is DenseNet-121 [4]. The last fully connected layer in the DenseNet is changed into a 1024-dimensional feature extractor layer. The stride in the last pooling layer is changed into 1 (identity layer). The whole network is trained with a cross-entropy/triplet combined loss. Then, we train another two models with either soft margin triplet loss or color jitters for model ensemble. After training three models, their features are first normalized, then concatenated, at last normalized again. The final feature is a 3072-dimensional normalized feature. The performance of this feature is investigated in Section 4.

Locality-aware threshold tuning. As shown in Fig. 6, the Euclidean distances between positive pairs are smaller, and the distances between negative pairs are larger. The average Euclidean distance between positive pairs is denoted as μ_p, and the average Euclidean distance between negative pairs is denoted as μ_n. In equation 2, the normalize parameter $norm = \frac{\mu_n - \mu_p}{2}$ is set as half of the distance between μ_p and μ_n.

The threshold, however, requires more tuning. First, we have a key finding. As shown in Fig. 6, positive pair distances are closer between within-camera pairs than cross-camera pairs. This is because the vehicle appearance changes continuously and subtly inside a camera. However, it tends to change more drastically cross cameras. In fact, the lighting condition, viewing angle, scale, and resolution all changes smoothly inside a camera, but more strongly across cameras.

Based on this finding, we set the threshold smaller so it is more sensitive to the within camera appearance variance. Traditionally, the threshold $norm = \frac{\mu_n + \mu_p}{2}$ is set as the median value between μ_p and μ_n for cross-camera pairs, for both single-camera tracking and cross-camera tracking. In the proposed system, threshold $norm$ for single-camera tracking and cross-camera tracking is set separately.

For single-camera tracking, the threshold $thres$ is set according to the average distance between single-camera data pairs. In fact, it is tuned to 0.68 for single camera tracking, which is the median value between μ_p and μ_n for within-camera pairs. For cross-camera tracking, the thresh-

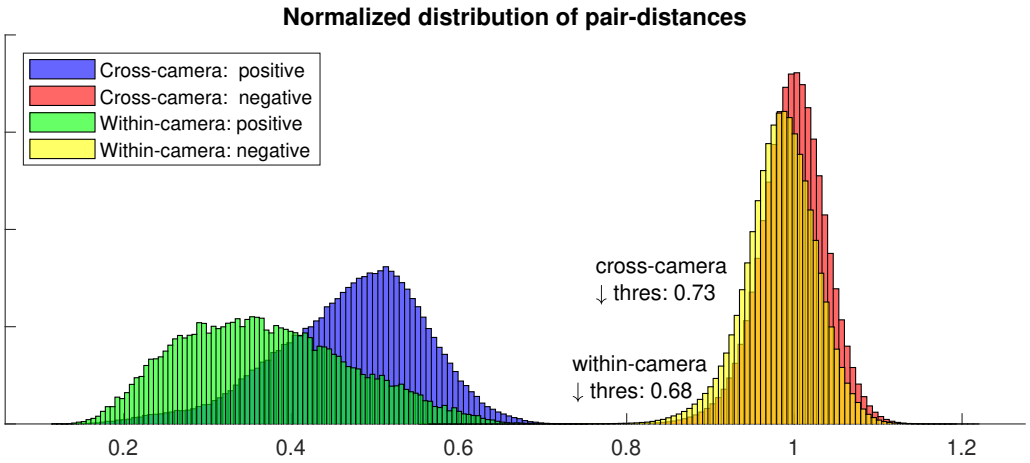


Figure 6: Euclidean distance between positive pairs and negative pairs. The pairwise distance between within-camera positive pairs are significantly smaller than that of cross-camera pairs. This is mainly due to small intra-camera lighting and perspective variation. In the proposed system, the threshold for positive/negative pair is also adjusted for single camera tracking.

Model	mAP (%)	rank-1 (%)
baseline	28.6	70.2
soft margin	29.1	70.5
color jitters	30.3	72.1
ensemble	**31.2**	**72.5**

Table 1: Vehicle re-ID performance on VeRi dataset.

old *thres* is set according to the average distance between cross-camera data pairs, which is the same as traditional settings. This results in a larger threshold at 0.73.

4. Experiment

4.1. Dataset and Evaluation Protocol

Dataset. The AI-City 2019 challenge MTMCT (track-1) dataset contains videos for urban intersections and highway. It contains a total of 40 cameras, in 5 scenarios. Scenarios S01, S03, S04 are used for training. S02 and S05 are used for testing. Videos are 960p or better, and most have been captured at 10 frames per second. There are 195.03 minutes of videos in total. The combined length of the training videos is 58.43 minutes, and the testing videos 136.60 minutes. In the MTMCT dataset, only cross-camera identities (vehicles appeared in multiple cameras) are labeled.

For re-ID feature training, the re-ID dataset in AI-City 2019 challenge (track-2) is used. There are 36935 images from 333 vehicles in the training set. For re-ID feature testing, we use the query/gallery in the VeRi dataset [9, 7].

Evaluation protocol. For MTMCT, multi-camera tracking (MCT) IDF-1 score are considered following [15]. In fact, the AI-City MTMCT dataset only consider MCT performance. For re-ID, we use mean average precision (mAP) and rank-1 precision following [25].

Name	IDF1	IDP	IDR
0.67_nms	0.6519	0.6300	0.6770

Table 2: Detailed results on the test set.

4.2. Implementation Details

For MTMCT, tracklets size is set to 10 frames. Next, a 30-frame-long sliding window is used to merge tracklets into trajectories. Following that, a 500-frame-long window is used in each sub-scene to create cross-camera identities. At last, for S05, all the cross-camera identities are merged together with a 2,400-frame-long window.

Note that the detection bounding boxes are enlarged by 40 pixels in both height and width, following the dataset settings. Then, the re-ID feature is extracted based on the enlarged bounding boxes. Detection bounding boxes with area smaller 1,000 pixels square are excluded from the system.

Moreover, after the formulation of the trajectories, we remove some overlapped bounding boxes through non-maximum-suppression (NMS). The NMS in our system follows two rules. First, the bounding box closer to the camera is kept. Second, only the bounding boxes with more than 2/3 area overlapped are removed.

Besides, the projection from the camera plane onto the map plane is carried out based on the provided homography matrix. These homography matrices are generated based on

the longitude and latitude, plus the position in the camera plane of the key point.

For re-ID feature training, the batch size is set to 64. As for triplet settings, images per identity is set to 4, and the margin is set to 0.3. Label smooth techniques [17] is also applied. The input size is set to 256×256. Following [11], we warm up the model linearly for 10 epochs, and train for 120 epochs. The learning rate is set to 0.01, and decays $0.1\times$ after the $30, 60, 80$ epochs.

All the experiments are conducted on a server with 6-core Intel Xeon processor, two NVIDIA 1080ti GPUs.

4.3. Evaluation

Vehicle re-ID. The performance of the proposed re-ID feature is shown in Table 1. All the models are trained on the re-ID dataset of the AI-City challenge (track-2). The baseline model with cross-entropy (with label smooth) and triplet loss perform decently on the VeRi dataset. With the inclusion of soft margin and color jitters, the performance gradually increases. At last, the 3072-dimensional feature in the ensemble model provides the best performance on the VeRi dataset.

Rank	Team ID	IDF Score
1	21	0.7059
2	49	0.6865
3	12	0.6653
4	53	0.6644
5	**97 (Ours)**	**0.6519**
6	59	0.5987
7	36	0.4924
8	107	0.4504
9	104	0.3369
10	52	0.2850

Table 3: Leader board on AI-City challenge for city-scale multi-camera vehicle tracking.

Vehicle MTMCT. For vehicle MTMCT, we only have access to the online testing set, which has a maximum submission count of 20. We report the top performing MTMCT results of the proposed system in Table 2 and Table 3. Overall, the system achieved 5^{th} place in the 2019 AI-City vehicle MTMCT challenge.

5. Conclusion

In this paper, multiple cues including ROI refinement, trajectory smoothness, and sub-scene division are employed to minimize the candidate pool for the city-scale vehicle MTMCT system. To best fit the reduced candidate pool, a more acute threshold is set in the similarity estimation module. Based on only the provided detection/training data, we achieved 5^{th} performance in the 2019 AI-City MTMCT challenge. In the future, we will continue to investigate the application and usage of the vehicle trajectories on the map plane.

References

[1] Ming-Ching Chang, Yi Wei, Nenghui Song, and Siwei Lyu. Video analytics in smart transportation for the aic'18 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 61–68, 2018.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[3] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. Challenges on large scale surveillance video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 69–76, 2018.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[5] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

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

[7] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[9] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, pages 869–884. Springer, 2016.

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

[11] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bags of tricks and a strong baseline for deep person re-identification. *arXiv preprint arXiv:1903.07071*, 2019.

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

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

[14] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, et al. The 2018 nvidia ai city challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–60, 2018.

[15] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[16] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6036–6046, 2018.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

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

[19] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David C. Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *CVPR 2019: IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–115, 2018.

[21] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. *arXiv preprint arXiv:1811.07258*, 2018.

[22] Chih-Wei Wu, Chih-Ting Liu, Cheng-En Chiang, Wei-Chih Tu, and Shao-Yi Chien. Vehicle re-identification with the space-time prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 121–128, 2018.

[23] Kwangjin Yoon, Young-min Song, and Moongu Jeon. Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views. *IET Image Processing*, 12(7):1175–1184, 2018.

[24] Zhimeng Zhang, Jianan Wu, Xuan Zhang, and Chi Zhang. Multi-target, multi-camera tracking by hierarchical clustering: recent progress on dukemtmc project. *arXiv preprint arXiv:1712.09531*, 2017.

[25] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.