

# Synthehicle: Multi-Vehicle Multi-Camera Tracking in Virtual Cities

Fabian Herzog Junpeng Chen Torben Teepe Johannes Gilg  
 Stefan Hörmann Gerhard Rigoll  
 Technical University of Munich, Germany

## Abstract

Smart City applications such as intelligent traffic routing, accident prevention or vehicle surveillance rely on computer vision methods for exact vehicle localization and tracking. Privacy issues make collecting real data difficult, and labeling data is a time-consuming and costly process. Due to the scarcity of accurately labeled data, detecting and tracking vehicles in 3D from multiple cameras proves challenging to explore. We present a massive synthetic dataset for multiple vehicle tracking and segmentation in multiple overlapping and non-overlapping camera views. Unlike existing datasets, which only provide tracking ground truth for 2D bounding boxes, our dataset additionally contains perfect labels for 3D bounding boxes in camera- and world coordinates, depth estimation, and instance, semantic and panoptic segmentation. The dataset consists of 17 hours of labeled video material, recorded from 340 cameras in 64 diverse day, rain, dawn, and night scenes, making it the most extensive dataset for multi-target multi-camera tracking so far. We provide baselines for detection, vehicle re-identification, and single- and multi-camera tracking. Code and data are publicly available.<sup>1</sup>

## 1. Introduction

As cities grow larger in population, increased car traffic causes jams, pollution, and accidents. Future smart cities, where multiple sensors (e.g., RGB cameras) are placed near crossroads, could reduce these problems through intelligent traffic management driven by computer vision.

In particular, multi-target multi-camera tracking (MTMCT) is an essential task for such methods as it enables 3D localization and provides information for scene understanding. In MTMCT, distinct objects must be unambiguously tracked across multiple cameras through space and time. The tracking problem is generally challenging, even in the single-camera case, primarily due to occlusions. Information added by multiple cameras can

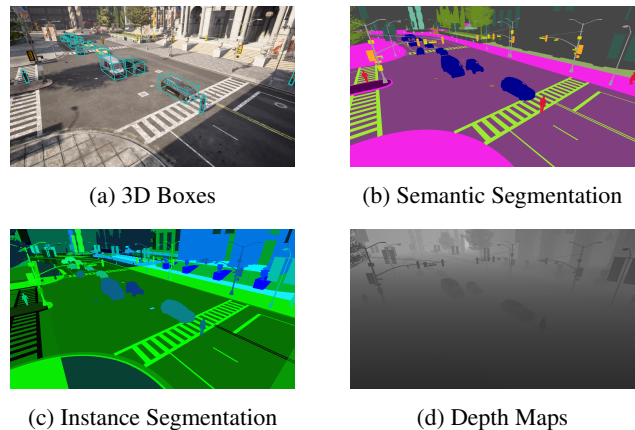


Figure 1: **Ground truth annotation in Synthehicle.** The proposed dataset contains perfect ground truth annotations for 3D detection (a), semantic (b), instance and panoptic (c) segmentation and depth estimation (d).

be beneficial but complicates data processing. Existing solutions for MTMCT [52, 24, 48, 39, 44] are mainly based on tracking-by-detection, in which an object detector is applied to obtain frame-wise sets of 2D bounding boxes which are then processed by a deep neural feature extractor for data association. Recently, the single-camera tracking community has started paying more attention to more complex tasks, such as 3D multi-object tracking [63, 69] and tracking and segmentation [59, 62]. Data annotation, particularly for multi-camera setups and 3D localization and segmentation, is time-consuming and expensive. To this day, there is no real dataset for multi-target multi-camera tracking with 3D and segmentation annotations. Both enable potentially better 3D localization than approaches based on 2D bounding boxes.

We address the scarcity of 3D and segmentation data and introduce *Synthehicle*, a massive synthetic dataset for multiple vehicle detection, tracking, and segmentation across multiple cameras with overlapping and non-overlapping field of views (see Figure 1 for example annotations). The dataset consists of 17 hours of labeled video material,

<sup>1</sup>Code and data: <https://github.com/fubel/synthehicle>

recorded from 340 cameras placed around crossroads and highways in 64 diverse day, rain, dawn and night scenes, and has been created using CARLA [12]. Summarized, our contributions are:

- a massive synthetic dataset for multi-target multi-camera tracking with 2D, 3D, segmentation and depth annotations;
- a public evaluation server to test methods against ground truth of the test sets; and
- baseline results for 2D detection, vehicle re-identification, single- and multi-camera multi-vehicle tracking tasks.

Code for data generation, detection, re-identification and tracking, as well as all the generated data and evaluation scripts are publicly available.

## 2. Related Work

**Vehicle Detection and Tracking Datasets** CityFlow [56, 46] is the established dataset to train and test vision methods for smart city applications such as multi-camera vehicle tracking, vehicle re-identification, and vehicle counting. However, it does not include 3D box annotations, depth maps, or segmentations. KITTI [19, 18] is loosely related as it includes data for 2D and 3D detection and tracking of vehicles. As KITTI aims at improving autonomous vehicles, the data is captured from specific ego-motion camera angles. nuScenes [4], also designed for autonomous driving, is similar to KITTI, but significantly larger. It provides more ego-motion camera angles and includes night and rain scenarios. Waymo Open [55] is similar in design but even more extensive than nuScenes. Neither KITTI, nuScenes nor Waymo Open provide the data for smart city scenarios. They focus on ego-motion and stereo vision by design and are unsuitable for studying problems such as vehicle re-identification and multi-vehicle multi-camera tracking.

**Person Tracking Datasets** Many datasets have been proposed for the related tasks of single- and multi-camera *person* tracking. For single-camera scenarios, the MOT challenges [41, 9, 60] are the established benchmarks to compare person trackers. The PETS09 [15] dataset can be considered the first relevant dataset for multi-camera person tracking. Other and larger datasets in this area are the EPFL-RLC [6], CAMPUS [65], MCT [8] datasets. WILDTRACK [5] is an extensive HD dataset developed for multiview detection and tracking, among other things. The DukeMTMC dataset, the largest real multi-camera tracking dataset, is no longer available due to privacy issues [50, 21]. To this day, there is no real dataset to substitute it. Recent trends in single-camera person tracking are based on 3D detections [63, 69] and tracking and segmentation [59, 62], where the objects of interest have to be tracked pixel-wise.

Both tasks allow a more exact localization of objects. Synthehicle is the first dataset to include 3D and segmentation ground truth for multi-camera multi-vehicle tracking.

**Vehicle Re-Identification Datasets** Just like person tracking relies on person feature extraction, most multi-vehicle trackers build upon vehicle re-identification methods [48, 39, 57, 54, 67, 32, 66], which are usually trained on the VeRi-776 [36, 34, 37] or CityFlow re-identification [56] datasets. Compared to VeRi-776, CityFlow was recorded in more diverse scenarios and viewing angles. Because trackers have to deal with occlusions and bad lighting conditions, the VERI-Wild vehicle re-identification dataset [38] has been proposed to provide images and annotations recorded in the wild.

**Synthetic Datasets** Recording and labeling real data is time-consuming and expensive. In addition, data protection rights are potentially infringed when recording humans. Therefore, numerous synthetic datasets have been published to solve common computer vision tasks, such as segmentation [58, 20, 53, 27, 49, 31], detection [40, 1], pose estimation [14], and tracking [14, 30, 13, 16, 10, 26]. In person tracking, the most significant synthetic datasets are JTA [14], which presents a dataset for single-camera pose-tracking based on the video game GTA V. Closely related, MTA [30] offers multi-camera multi-person tracking data that can be seen as a replacement for the DukeMTMC dataset. MOTSynth showed that training on synthetic data can improve tracking results on real data [13]. The synthetic VehicleX [68] dataset was generated for vehicle re-identification. Related datasets also based on CARLA are KITTI-CARLA [10], which provides synthetic lidar data analogous to KITTI [19], V2I-CARLA [61] for synthetic vehicle re-identification, and the Paris-CARLA-3D [11] dataset for 3D mapping.

## 3. Dataset Analysis

### 3.1. Overview and Comparison

So far, the MTMC vehicle tracking community has focused on *tracking-by-detection*, a methodology based on 2D object detections and subsequent appearance feature extraction [48, 39, 57, 54, 67, 32, 66]. Unlike other datasets, which are designed for tracking targets enclosed by 2D bounding boxes, Synthehicle provides 3D annotations in terms of 3D bounding boxes and world coordinates, segmentations (instance, semantic and panoptic), and depth images. In Table 1, we compare Synthehicle to established tracking datasets, such as PETS09, WILDTRACK and CityFlow. Synthehicle is the only dataset with all of the listed annotation types, and the longest dataset in duration.



Figure 2: **Ambience configurations.** In Synthehicle, scenes are rendered under four different ambience configurations: day (a), dawn (b), rain (c), and night (d).

	Dataset	# Cams	# Boxes	# Scenes	Density	Duration	Targets	3D Boxes	Depth	Segmentation
Real	DukeMTMC (offline) [51]	8	4 077 132	1	1.9	11.33h	Person	✗	✗	✗
	CAMPUS [65]	16	12 264	4	–	45m	Person	✗	✗	✗
	EPFL-RLC	3	6,132	1	–	6.6 min	Person	✗	✗	✗
	PETS09	7	4650	8	–	2 min	Person	✗	✗	✗
	WILDTRACK [5]	7	56 000	1	23.8	1h	Person	✗	✗	✗
	CityFlow [56]	46	313 931	6	2.5	3.58h	Vehicles	✗	✗	✗
Synth.	JTA [14]	1*	10 000 000	512	20	4.27h	Person	✗	✗	✗
	MOTSynth [13]	1*	40 780 800	768	29.5	15.36h	Person	✗	✓	✓
	MTA [30]	6	37 324 348	1	24.8	10.2h	Person	✗	✗	✗
	<b>Synthehicle (ours)</b>	<b>340</b> × 7.39	<b>4 623 184</b> × 14.72	<b>64</b> × 10.67	<b>7.45</b> × 2.98	<b>17.00h</b> × 4.74	Vehicles	✓	✓	✓

Table 1: **Overview of existing MTMCT datasets.** The table shows a comparison of Synthehicle to related datasets. A ✓ indicates whether 3D boxes, depth annotations and segmentations are included in the ground truth, respectively. For multi-camera datasets, the number of cameras corresponds to the total number of videos in the dataset. Naturally, single-camera datasets have one camera per scene, and their camera numbers are marked with \*. The last row compares the individual statistics of our dataset directly to CityFlow.

Compared to the closest related real dataset, CityFlow, Synthehicle is three times denser in terms of average number of vehicles per frame, has more than ten times as many scenes and fourteen times as many annotated bounding boxes. It also surpasses all other multi-camera tracking datasets in the number of cameras. Table 2 provides a detailed list of all Synthehicle scenes and their classification into train and test split.

### 3.2. Data Recording

CARLA [12] is an open-source simulation tool for building urban traffic scenarios and has been successfully employed to generate numerous synthetic datasets for computer vision tasks [58, 20, 53, 27, 49, 31, 40, 1, 14, 30, 13, 16, 10, 26]. We utilize CARLA’s rich set of realistic simulated sensors to render urban vehicle tracking scenarios using RGB, depth, and semantic LIDAR sensors in CARLA’s eight pre-designed town maps. The data recording process can be described as follows.

First, we define two scenes for each of the eight towns - one for an overlapping camera view setup (O) and the other for a non-overlapping setup (N). We place a varying number of cameras (3 to 8) in each scene. The cameras are

placed such that the vehicles are viewed from a high angle to mimic real-world positions, e.g., on top of traffic lights, similarly to CityFlow [56]. After defining the camera networks, we randomly spawn vehicles and pedestrians. The maximum number of vehicles that could spawn on a map is set to 200. Pedestrians mainly fulfill the task of influencing the otherwise monotonous flow of traffic. Models and appearances for vehicles and pedestrians are randomly chosen from CARLA’s model pool and a list of realistic vehicle colors, matching real-life urban scenarios (cf. Figure 4).

Vehicle routing and rules are controlled by CARLA’s *TrafficManager*, which was designed to manage vehicles in autopilot mode. Some vehicles will disobey traffic rules by driving too fast or crossing red traffic lights, providing more diverse and less predictable trajectories. For every defined camera, we deploy an RGB sensor to record the traffic scenes with a resolution of  $1920 \times 1080$  px and a frame rate of 10 fps. Semantic rotating LIDAR sensors capture information about all vehicles in a camera’s field of view. The sensor is implemented using ray-casting and exposes all information about objects hit by a ray. We use the sensor to filter out annotations for heavily occluded objects, e.g., for cars hidden behind buildings. We attach two different seg-

Scene	# Cams	# Boxes	# Tracks	Density	Test
Town01-O-dawn	4	48 048	82	6.67	✗
Town01-O-day	4	57 539	80	7.99	✗
Town01-O-night	4	51 725	73	7.18	✗
Town01-O-rain	4	70 595	69	9.80	✗
Town02-O-dawn	3	61 337	69	11.35	✗
Town02-O-day	3	64 093	74	11.86	✗
Town02-O-night	3	47 619	64	8.81	✗
Town02-O-rain	3	43 416	72	8.04	✗
Town03-O-dawn	8	200 088	126	13.89	✗
Town03-O-day	8	192 994	124	13.40	✗
Town03-O-night	8	207 035	117	14.37	✗
Town03-O-rain	8	192 072	115	13.33	✗
Town04-O-dawn	4	33 644	39	4.67	✗
Town04-O-day	4	35 816	47	4.97	✗
Town04-O-night	4	28 173	44	3.91	✗
Town04-O-rain	4	23 086	42	3.20	✗
Town05-O-dawn	6	13 1720	96	12.19	✗
Town05-O-day	6	11 3413	96	10.50	✗
Town05-O-night	6	10 7870	88	9.98	✗
Town05-O-rain	6	12 7385	96	11.79	✗
Town06-O-dawn	4	19 734	45	2.74	✓
Town06-O-day	4	13 918	55	1.93	✓
Town06-O-night	4	15 859	45	2.20	✓
Town06-O-rain	4	19 308	57	2.68	✓
Town07-O-dawn	4	54 679	58	7.59	✓
Town07-O-day	4	56 314	57	7.82	✓
Town07-O-night	4	80 797	46	11.22	✓
Town07-O-rain	4	46 584	59	6.47	✓
Town10HD-O-dawn	5	95 426	110	10.60	✓
Town10HD-O-day	5	16 5259	116	18.36	✓
Town10HD-O-night	5	89 170	100	9.90	✓
Town10HD-O-rain	5	98 855	125	10.98	✓

Scene	# Cams	# Boxes	# Tracks	Density	Test
Town01-N-dawn	6	85 515	143	7.91	✗
Town01-N-day	6	77 372	140	7.16	✗
Town01-N-night	6	72 420	140	6.70	✗
Town01-N-rain	6	61 648	130	5.70	✗
Town02-N-dawn	5	50 676	78	5.63	✗
Town02-N-day	5	60 452	83	6.71	✗
Town02-N-night	5	59 928	86	6.65	✗
Town02-N-rain	5	56 781	95	6.30	✗
Town03-N-dawn	5	92 879	182	10.31	✗
Town03-N-day	5	79 180	172	8.79	✗
Town03-N-night	5	70 648	166	7.84	✗
Town03-N-rain	5	67 393	157	7.48	✗
Town04-N-dawn	5	53 505	149	5.94	✗
Town04-N-day	5	48 521	134	5.39	✗
Town04-N-night	5	52 177	150	5.79	✗
Town04-N-rain	5	56 749	161	6.30	✗
Town05-N-dawn	5	59 804	131	6.64	✗
Town05-N-day	5	61 488	130	6.83	✗
Town05-N-night	5	65 224	142	7.24	✗
Town05-N-rain	5	56 459	122	6.27	✗
Town06-N-dawn	7	41 687	186	3.30	✓
Town06-N-day	7	39 087	188	3.10	✓
Town06-N-night	7	37 006	189	2.93	✓
Town06-N-rain	7	44 945	184	3.56	✓
Town07-N-dawn	7	13 936	42	1.10	✓
Town07-N-day	7	17 585	43	1.39	✓
Town07-N-night	7	21 916	56	1.73	✓
Town07-N-rain	7	12 354	41	0.98	✓
Town10HD-N-dawn	7	134 134	150	10.64	✓
Town10HD-N-day	7	122 552	149	9.72	✓
Town10HD-N-night	7	136 116	137	10.80	✓
Town10HD-N-rain	7	119 476	144	9.48	✓

Table 2: **Overview over all 64 Synthehicle scenes.** Scenes have a frame rate of 10fps, a resolution of 1920x1080 and a duration of 1800 frames per camera. The dataset is split into train and test scenes. The markers -O- and -N- in the scene names indicate overlapping and non-overlapping camera topology, respectively.

mentation sensors and a depth sensor to each RGB camera to record semantic and instance segmentations and to obtain depth information. The recording of all sensors lasts 1800 frames. Scenes are recorded as described above in four different ambience configurations: Day, dawn, rain, and night. Figure 2 shows an example frame recorded from the same camera under different configurations. When recording a scene for a weather configuration, all vehicle and pedestrian spawns and traffic flows will be randomized - only the sensor placement is constant for each scene. This way, Synthehicle provides a vast variety of traffic flows and vehicle trajectories. Finally, we extract perfect ground truth annotations while recording the scenes.

### 3.3. Data Types

We use CARLA’s Python API to extract a variety of ground truth annotations: Camera calibrations, 2D and 3D detections, semantic, instance and panoptic segmentations, depth information and multi-camera tracking ground truth.

**Calibrations** For each camera, we obtain its  $3 \times 3$  camera intrinsic matrix  $\mathbf{K}$  and its  $4 \times 4$  world-to-camera matrix  $\mathbf{M}_{w2c} = [\mathbf{R}, \mathbf{t}]$ , which is the inverse of the camera extrinsic matrix. These matrices can be used for converting world points to image points and vice versa. We also provide  $(x, y, z)$  3D world positions for all cameras and their pitch, roll and yaw.

**3D Detections** For each object in the scene, we obtain its 3D bounding box in world  $(x, y, z)$ -coordinates directly from CARLA. The oriented bounding box is defined by eight corner points and yaw rotation. Each world point  $(x_w, y_w, z_w)^T \in \mathbb{R}^3$  is projected to camera coordinates using the  $4 \times 4$  world-to-camera matrix  $\mathbf{M}_{w2c}$  to obtain the point in image coordinates  $(x, y)^T$  via

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \\ 1 \end{bmatrix} = \mathbf{M}_{w2c} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \tilde{y}_1 \\ -\tilde{z}_1 \\ \tilde{x}_1 \end{bmatrix}, \quad (1)$$



where  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  denotes the camera intrinsic matrix and  $(x, y)^T = (x_I/z_I, y_I/z_I)^T$ . Note that CARLA uses UnrealEngine’s left-handed  $z$ -up coordinate system, which is why the axes have to be permuted and the  $z$ -axis reversed before multiplying with  $\mathbf{K}$  to obtain  $(x, y)^T$  in image space. We store 3D boxes in COCO-format [33] as  $\{(x_i, y_i, z_i, w_i, h_i, l_i, \theta_i)\}_i$ , where  $(x_i, y_i, z_i)$  is the 3D center point,  $(w_i, h_i, l_i)$  describe weight, height and length of the box, and  $\theta_i$  is the yaw rotation.

**2D Detections** The 2D detections are obtained in image coordinates from the projected 3D detections. If  $B_{3D} = \{c_1, \dots, c_8\}$  is the 3D bounding box with its eight corner points, we obtain  $(x_1, y_1, x_2, y_2)^T = (\min_x B_{3D}, \min_y B_{3D}, \max_x B_{3D}, \max_y B_{3D})^T$ , i.e., by choosing the 2D box as the smallest rectangle containing all projected 3D vertices. Since 2D boxes acquired from this min-max-projection are usually larger than the enclosed target object, we tighten the box using the objects semantic segmentation label (see Figure 3). Bounding boxes smaller than  $32 \times 32$  are filtered out from 2D detections, but are still kept in our general annotations.

**Segmentations** CARLA provides two sensors for segmentation: A semantic segmentation and an instance segmentation camera. We capture images from both cameras for each frame and directly store the corresponding output images. Having obtained semantic and instance segmentation labels, the panoptic pixel labels can easily be calculated by looking up the corresponding segmentation pixel for each instance segmentation pixel. Figure 1 illustrates an example of semantic and instance segmentation annotations.

**Depth Buffer** Leveraging CARLA’s depth sensor we also capture the depth map for each frame. The depth information can be used to improve camera projections since it provides scaling in depth dimension. Figure 1 includes an example.

**Tracking** We store the tracking ground truth for 2D and 3D boxes in COCO and MOTChallenge [41] format. Unlike CityFlow [56], our ground truth contains boxes for both vehicles and pedestrians, and boxes are included also if they are only visible in one camera.

## 4. Experiments

We conduct the following experiments to show the performance of existing methods on the Synthechicle dataset and to provide baselines for future research: 2D vehicle detection (Section 4.1), vehicle re-identification (Section

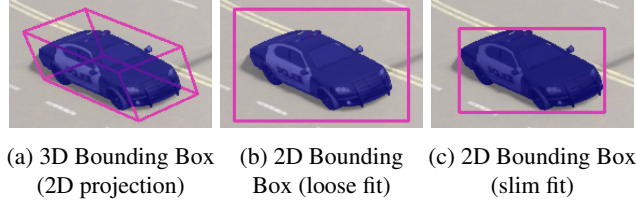


Figure 3: **Bounding Boxes in Synthechicle.** The initial 3D bounding boxes (a) are transformed into 2D boxes (b) by a min-max procedure. Using semantic segmentation, this box is then refined to wrap tightly around the target object.



Figure 4: **Vehicle colors in CARLA.** The top two rows show vehicles from CARLA in the realistic color palette. The last row shows vehicles in a customized color palette. To increase difficulty of re-identification tasks, we only use realistic colors in Synthechicle.

Train	Test	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
–	All	0.242	0.438	0.217	0.003	0.149	0.460
All	All	0.597	0.842	0.651	0.151	0.480	0.785
Day	Day	0.587	0.842	0.652	0.082	0.469	0.769
Dawn	Dawn	0.608	0.866	0.660	0.139	0.505	0.792
Rain	Rain	0.568	0.822	0.611	0.122	0.447	0.791
Night	Night	0.506	0.780	0.540	0.119	0.381	0.666
All	Day	0.626	0.870	0.693	0.116	0.510	0.808
All	Dawn	0.640	0.882	0.701	0.182	0.533	0.827
All	Rain	0.597	0.840	0.648	0.162	0.476	0.818
All	Night	0.522	0.777	0.560	0.171	0.383	0.696

Table 3: **2D detection performance.** We evaluate the YOLOX-x object detector with pretrained COCO weights under different train-test-split configurations. Values are given in %.

4.2), single-camera multi-vehicle tracking (Section 4.3), and multi-vehicle multi-camera tracking (Subsection 4.4).

### 4.1. 2D Vehicle Detection

An object detector processes an image  $I \in \mathbb{R}^{W \times H \times 3}$  and returns a set of distinct objects  $\{(x_i, y_i, w_i, h_i, c_i, s_i)\}_i$  with localization coordinates

$(x_i, y_i, w_i, h_i)$ , optionally with class  $c_i$  and confidence score  $s_i$ . As such, object detectors are an important part of most tracking pipelines, which extract features from detected objects to match them accordingly. For smart cities, 2D vehicle detection is important for vehicle localization.

**Experimental Setup** We use the YOLOX-x model [17] in the mmdetection [7] framework. Research on training detectors on synthetic data suggest that variety is more important than data size [13]. Therefore, for training and testing, we sample every tenth frame from all of our respective scenes and include it in the corresponding detection split.

**Results** Table 3 lists the results of our detection training. When using pretrained COCO weights only and thereby not training on Synthehicle at all, we obtain mediocre detection results. By visualizing false positives and negatives, we observe that YOLOX-x struggles to detect special CARLA vehicles, such as fire engines and cybertrucks. However, when fine-tuning on the Synthehicle train splits, performance increases significantly. As expected, performance on rain and night scenes is worse than on day and dawn scenes. Interestingly, we obtain the highest performance on dawn scenes. This might be because in dawn scenarios there are almost no shadows present in the scenes, potentially reducing the number of false positives. Compared to training and testing on weather-specific train and test splits, training on all splits and then testing on the individual test splits leads to increased performance. Thus, the variety within the training data is crucial for a well-generalizing model.

## 4.2. Vehicle Re-Identification

Object re-identification is an image retrieval problem in which a gallery set of  $m$  images  $\mathcal{G} = \{\mathbf{I}_{G,i}\}_{i=1}^m, \mathbf{I}_{G,i} \in \mathbb{R}^{W \times H \times 3}$ , is ranked by similarity to a query image  $\mathbf{I}_Q \in \mathbb{R}^{W \times H \times 3}$ , with the goal that the most similar gallery image belongs to the same identity (i.e., class) as the query image. In particular, images in object re-identification are obtained from multiple distinct cameras. The desired similarity measure in question is usually obtained by extracting features using a deep neural network and measuring their distances [28, 70, 46, 45, 25].

Dataset	Training	Gallery	Query	# Classes
Synthehicle (day)	24 509	11 556	2 889	223
Synthehicle (dawn)	23 963	8 039	2 010	220
Synthehicle (rain)	22 171	7 036	1 760	205
Synthehicle (night)	22 943	8 775	2 194	205
Synthehicle (all)	93 586	35 407	8 852	853

Table 4: **Synthehicle Re-Identification Splits.** We create five different splits for re-identification training.

Person and vehicle re-identification have received much attention in the past. Both tasks have to deal with similar problems like changing camera angles and lighting conditions. However, while people look and dress somewhat differently, vehicles can appear similarly. Additionally, vehicles provide almost no spatial-temporal pose information since, unlike humans, their shapes remain almost unchanged under movement. Important real datasets for vehicle re-identification are the VeRi dataset [35] with 776 vehicles captured over 20 cameras, and the AICity ReID dataset [56, 43], which is provided as part of the AICity-Challenge.

**Experimental Setup** We train vehicle re-identification on different train and test splits of Synthehicle using the fastreid [23] framework with a ResNet50 backbone [22] and instance normalization [47]. Train and test splits were created by sampling every tenth frame and cropping detections using ground truth annotations. Crops smaller than  $50 \times 50$  were filtered out. The vehicle images are scaled to  $256 \times 256$  pixels for the training and deformed if their bounding box is not quadratic. We always train up to 140 epochs with early stopping using the Adam optimizer [29]. During testing, we normalize the features in the Euclidean norm and use the cosine distance between feature vectors to calculate the cumulative matching characteristics (CMC) [42]. We report the CMC rank-1 accuracy (r1) and mean Average Precision (mAP) for evaluation. As in training, the images are scaled to  $256 \times 256$  for testing.

**Results** Table 5 lists the results produced by the fastreid model with different train and test combinations. As expected, re-identification works best in the day and dawn scenes and performs slightly worse in the rain and night scenes. If trained on all scenes, performance is increased on all test splits. This is surprising since CARLA only provides a limited number of vehicle models, and many spawned objects will be identical in appearance, but different in vehicle ID. The results indicate that the “more data is better”-assumption of deep learning also holds in this particular case.

## 4.3. Single-Camera Vehicle Tracking

In single-camera multi-object tracking-by-detection, the task is to generate consistent tracks from a time-ordered set of images  $\mathcal{I} = \{\mathbf{I}_t\}_t$  and corresponding object detections  $\mathcal{D} = \{D_t\}_t, D_t = \{(x_i, y_i, w_i, h_i)_i\}_t$  (cf. Section 4.1), such that every distinct target is assigned an unambiguous and unique ID across all time frames. Occlusions, lighting variations, and false positive or negative detections make tracking challenging. Important trackers include, among others, DeepSORT [64], Tracktor [2], and CenterTrack [71].

Train	Test	mAP (%)	rank 1 (%)	rank 5 (%)	rank 10 (%)
All	All	47.82	51.55	72.29	80.03
Day	Day	59.89	62.02	77.12	85.04
Dawn	Dawn	47.57	51.17	72.30	80.47
Rain	Rain	39.08	48.44	72.31	80.29
Night	Night	27.04	36.84	58.76	67.41
All	Day	60.38	58.50	78.56	84.58
All	Dawn	53.21	55.75	82.51	53.21
All	Rain	45.33	52.60	75.66	83.76
All	Night	32.69	37.25	60.78	68.88

Table 5: **Vehicle Re-Identification Performance.** The table shows a performance comparison under different train and test splits.

**Experimental Setup** We use the DeepSORT [64] tracker with the YOLOX-x models trained in Section 4.1 and the re-identification weights obtained from the experiments in Section 4.2. We follow the evaluation protocols of single-camera tracking by utilizing the CLEAR MOT metrics [3], with multiple-object tracking accuracy and precision (MOTA and MOTP), and number of ground truth (GT), mostly tracked (MT), partially tracked (PT) and mostly lost (ML) targets. We set the minimum detection height and width to  $32 \times 32$  to filter small false positive detections and set the detection confidence threshold to 0.3. The DeepSORT parameter `nn_budget`, which decides how many appearance features are considered for a track, is set to 100.

**Results** Table 6 shows the results of DeepSORT on all test scenes. Note that results are averaged over the respective number of camera videos included in a scene for performance metrics. As expected, tracking in day and dawn scenes generally yields superior performance than tracking on rain and night scenes. However, for Town06-O-night, DeepSORT performs better than in other scenes with the same camera setup. We conjecture that the 10 fps frame rate is sufficiently large to track objects in a scene with comparatively low density (cf. Table 2). Note also that scenes with different ambience configurations are rendered with different vehicle spawns and number of vehicles each. Only the camera setup is fixed. Overall, DeepSORT-based single-camera tracking on Synthehicle generates satisfying baseline results.

#### 4.4. Multi-Vehicle Multi-Camera Tracking

MTMCT is the extension of single-camera tracking to a multiple-camera setup, where the input set of images now comes from  $K$  distinct views  $\mathcal{I} = (\{\mathbf{I}_t^{(c_1)}\}_t, \dots, \{\mathbf{I}_t^{(c_K)}\}_t)$ , and the detections are obtained accordingly as  $\mathcal{D} = \{\mathcal{D}^{(c_i)}\}_{i=1, \dots, K}$ . Generally, there is no restriction on the camera topology and field of views can be overlapping or

non-overlapping. In practice, many MTMC trackers [52, 48] first apply a single-camera tracker on the individual camera videos to obtain single-camera results (i.e., tracklets)  $\mathcal{T}_1, \dots, \mathcal{T}_K$ , and then match those sets of local tracklets into a set of global tracklets  $\mathcal{T}$ .

**Method** We use a method based on ELECTRICITY [48], a 2020 challenge winner for the CityFlow dataset, with the single-camera tracklets  $\mathcal{T}_1, \dots, \mathcal{T}_K$  obtained by applying DeepSORT as in Section 4.3. For each tracklet, a query image is chosen, and all other images of that tracklet are selected to be gallery images. Using the feature extractor trained in Section 4.2, a query feature matrix  $\mathbf{Q} \in \mathbb{R}^{n \times f}$  and a gallery feature matrix  $\mathbf{G} \in \mathbb{R}^{m \times f}$  are built, where  $n$  is the number of tracklets,  $m$  is the number of gallery images, and  $f$  is the feature embedding size. After normalizing all query and probe features, i.e.  $\|\mathbf{q}_i\|_2 = 1 \forall i = 1, \dots, n$  and  $\|\mathbf{g}_j\|_2 = 1 \forall j = 1, \dots, m$ , we obtain the cosine distance matrix  $\mathbf{D}$  as

$$\mathbf{D} = \mathbf{Q}\mathbf{G}^T \in \mathbb{R}^{n \times m}. \quad (2)$$

The multi-camera tracking result is then derived from  $\mathbf{D}$  by merging the tracklets  $\mathcal{T}_i$  and  $\mathcal{T}_j$  if and only if both the feature distance between the query image of  $\mathcal{T}_i$  and the gallery images of  $\mathcal{T}_j$ , and the feature distance between the query image of  $\mathcal{T}_j$  and the gallery images of  $\mathcal{T}_i$  are below a specified threshold  $\theta$ , i.e.,

$$\mathbf{D}_{ij} < \theta \text{ and } \mathbf{D}_{ji} < \theta. \quad (3)$$

**Experimental Setup** We apply the method described above with  $\theta = 0.8$  on all test scenes. For evaluation, we report the IDF1, IDP and IDR metrics for multi-camera tracking as suggested by [50]. Features are extracted on single-camera tracklets obtained in Section 4.3 using the re-identification network trained in Section 4.2.

**Results** Table 6 list the performance of the multi-camera tracker on Synthehicle. Since the tracker is based on pre-extracted single-camera tracklets, there is a correlation between single-camera performance and multi-camera performance. The multi-camera tracker relies solely on re-identification and performance sometimes decreases in night scenes (e.g., in Town10HD-O-night). Figure 5 shows the influence of scene density and single-camera IDF1 performance on multi-camera IDF1 performance. Dense scenes increase the difficulty for our multi-camera tracking, since more gallery and query features have to be considered during matching, leading to potential association errors.

## 5. Conclusion

We have presented a massive synthetic dataset for tracking vehicles across multiple overlapping and non-

Scene	Single-Camera										Multi-Camera		
	MOTA	MOTP	IDF1	IDP	IDR	GT	MT	PT	ML	IDs	IDF1	IDP	IDR
Town06-N-day	54.3%	0.175%	65.8%	78.4%	56.8%	438	116	194	128	252	48.2%	58.1%	41.6%
Town06-N-dawn	55.0%	0.204%	64.4%	80.4%	53.7%	437	126	176	135	335	49.3%	61.8%	41.2%
Town06-N-rain	50.5%	0.191%	60.4%	77.2%	49.6%	474	124	204	146	389	41.4%	53.9%	34.0%
Town06-N-night	45.8%	0.183%	59.9%	76.2%	49.4%	455	105	187	163	257	39.6%	51.1%	32.7%
Town07-N-day	62.0%	0.157%	78.1%	79.3%	76.9%	142	77	59	6	35	46.6%	47.3%	45.9%
Town07-N-dawn	73.4%	0.173%	80.6%	90.3%	72.8%	141	68	66	7	46	59.2%	66.3%	53.4%
Town07-N-rain	62.3%	0.158%	78.4%	79.6%	77.2%	122	55	64	3	45	57.8%	59.2%	57.0%
Town07-N-night	50.6%	0.198%	62.4%	83.9%	49.7%	195	51	109	35	154	40.8%	54.8%	32.5%
Town10HD-N-day	53.8%	0.186%	65.9%	68.8%	63.2%	566	219	255	92	846	38.8%	40.9%	37.2%
Town10HD-N-dawn	54.3%	0.204%	63.7%	67.5%	60.2%	579	221	274	84	1157	36.3%	39.0%	34.3%
Town10HD-N-rain	50.8%	0.201%	63.7%	66.2%	61.5%	565	192	277	96	954	37.7%	40.5%	36.4%
Town10HD-N-night	54.8%	0.192%	68.6%	69.2%	68.0%	355	140	150	65	1004	50.8%	55.3%	50.4%
Town06-O-day	68.5%	0.145%	72.2%	85.5%	62.5%	150	47	50	53	35	46.7%	55.3%	40.4%
Town06-O-dawn	66.4%	0.147%	72.2%	87.4%	61.5%	143	53	58	32	53	45.4%	55.0%	38.7%
Town06-O-rain	58.6%	0.153%	68.3%	86.6%	56.3%	172	48	75	49	61	45.9%	60.5%	37.8%
Town06-O-night	69.7%	0.142%	72.4%	84.8%	63.2%	125	45	44	36	55	50.1%	58.8%	43.7%
Town07-O-day	71.5%	0.149%	72.7%	80.3%	66.3%	171	92	54	25	257	37.1%	41.0%	33.8%
Town07-O-dawn	74.7%	0.145%	77.4%	83.7%	72.0%	190	117	56	17	212	39.6%	43.4%	36.8%
Town07-O-rain	65.6%	0.156%	66.7%	75.2%	59.9%	172	77	67	28	256	42.6%	49.1%	38.3%
Town07-O-night	50.4%	0.213%	58.3%	62.7%	54.5%	127	56	46	25	1194	30.3%	32.9%	28.3%
Town10HD-O-day	61.9%	0.197%	71.2%	74.7%	68.0%	437	205	159	73	702	25.8%	27.1%	24.6%
Town10HD-O-dawn	64.9%	0.201%	68.0%	72.6%	63.9%	454	226	184	44	665	31.9%	34.9%	30.0%
Town10HD-O-rain	47.3%	0.212%	61.2%	68.7%	55.1%	519	194	239	86	740	30.9%	36.1%	27.8%
Town10HD-O-night	50.2%	0.220%	58.1%	61.5%	55.1%	431	163	230	38	908	24.5%	30.7%	23.3%
<b>Mean</b>	<b>59.0%</b>	<b>0.179%</b>	<b>67.9%</b>	<b>76.6%</b>	<b>61.5%</b>	<b>315</b>	<b>117</b>	<b>136</b>	<b>61</b>	<b>442</b>	<b>41.5%</b>	<b>48.0%</b>	<b>37.5%</b>

Table 6: **Single-Camera and Multi-Camera Vehicle Tracking Performance.** Performance of single- and multi-camera tracking on Synthehicle. Single-camera performance results are obtained for every individual camera in the scene, and then averaged for this table, while the discrete statistics (GT, ML, PT, ML, IDs) are aggregated by summation.

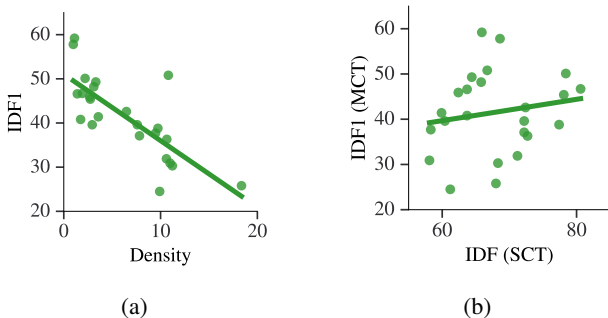


Figure 5: **Results analysis.** The scene density has a strong influence on multi-camera tracking performance (a). The quality of single-camera tracklets influences the multi-camera tracking performance, but less than expected (b).

overlapping cameras in various scenes with a wide variety of data annotations previously not included in similar datasets, such as semantic, instance and panoptic segmentations, depth maps, and over 4 million annotated 2D and 3D bounding boxes. With 17 hours of video material and 340 cameras, it is the largest available MTMCT dataset. The

ambience configurations included in our dataset allow for exploring multi-vehicle tracking under challenging conditions. We have demonstrated the performance of different baselines for vehicle detection, re-identification, and single- and multi-camera tracking tasks. Results on these tasks indicate that Synthehicle is a complex dataset with diverse and challenging scenarios. While the focus of our analysis has been MTMCT, our annotations can potentially enable the exploration of new tasks, such as 3D multi-target multi-camera tracking or multi-camera tracking and segmentation. Recording and labeling real datasets is a time-consuming task. Furthermore, in the case of large datasets, it cannot be assumed that persons contained in the dataset have given their consent. One-sided datasets could also reinforce problematic biases. Synthetic data, on the other hand, can be collected without any privacy violations and biases could potentially also be compensated for in such data.

**Acknowledgement** We gratefully acknowledge the financial support from Deutsche Forschungsgemeinschaft (DFG) under grant number RI 658/25-2.



## References

- [1] Giuseppe Amato, Luca Ciampi, Fabrizio Falchi, Claudio Gennaro, and Nicola Messina. Learning Pedestrian Detection from Virtual Worlds. In Elisa Ricci, Samuel Rota Bulò, Cees Snoek, Oswald Lanz, Stefano Messelodi, and Nicu Sebe, editors, *Image Analysis and Processing - ICIAP 2019 - 20th International Conference, Trento, Italy, September 9-13, 2019, Proceedings, Part I*, volume 11751 of *Lecture Notes in Computer Science*, pages 302–312. Springer, 2019.
- [2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking Without Bells and Whistles. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [3] Keni Bernardin and Rainer Stiefelwagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.
- [4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nusenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [5] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *CVPR*, pages 5030–5039, 2018.
- [6] Tatjana Chavdarova and François Fleuret. Deep multi-camera people detection. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 848–853. IEEE, 2017.
- [7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [8] W Chen, X Chen, and K Huang. Multi-Camera Object Tracking (MCT) Challenge, 2014.
- [9] 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[cs]*, Mar. 2020. arXiv: 2003.09003.
- [10] Jean-Emmanuel Deschaud. KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator. *arXiv preprint arXiv:2109.00892*, 2021.
- [11] Jean-Emmanuel Deschaud, David Duque, Jean Pierre Richa, Santiago Velasco-Forero, Beatriz Marcotegui, and François Goulette. Paris-CARLA-3D: A Real and Synthetic Outdoor Point Cloud Dataset for Challenging Tasks in 3D Mapping. *Remote Sensing*, 13(22):4713, 2021.
- [12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [13] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking? In *CVPR*, pages 10849–10859, 2021.
- [14] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, pages 430–446, 2018.
- [15] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE, 2009.
- [16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. *CoRR*, abs/1605.06457, 2016.
- [17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.
- [18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012.
- [20] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. *CoRR*, abs/1511.07041, 2015.
- [21] Adam Harvey and Jules LaPlace. Megapixels: Origins, ethics, and privacy implications of publicly available face recognition image datasets. *Megapixels*, 1(2):6, 2019.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [23] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020.
- [24] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-Target Multi-Camera Tracking by Tracklet-to-Target Assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020.
- [25] Fabian Herzog, Xunbo Ji, Torben Teepe, Stefan Hörmann, Johannes Gilg, and Gerhard Rigoll. Lightweight Multi-Branch Network For Person Re-Identification. In *2021 IEEE International Conference on Image Processing, ICIP 2021, Anchorage, AK, USA, September 19-22, 2021*, pages 1129–1133. IEEE, 2021.
- [26] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. Joint Monocular 3D Vehicle Detection and Tracking. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5389–5398. IEEE, 2019.

- [27] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G. Schwing. SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation - A Synthetic Dataset and Baselines. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3105–3115. Computer Vision Foundation / IEEE, 2019.
- [28] Su V Huynh. A strong baseline for vehicle re-identification. In *CVPR*, pages 4147–4154, 2021.
- [29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [30] Philipp Kohl, Andreas Specker, Arne Schumann, and Jürgen Beyerer. The MTA dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *CVPRW*, pages 1042–1043, 2020.
- [31] Philipp Krähenbühl. Free Supervision From Video Games. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2955–2964. Computer Vision Foundation / IEEE Computer Society, 2018.
- [32] Fei Li, Zhen Wang, Ding Nie, Shiyi Zhang, Xingqun Jiang, Xingxing Zhao, and Peng Hu. Multi-camera vehicle tracking system for AI City Challenge 2022. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3265–3273, 2022.
- [33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [34] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, WA, USA, July 11-15, 2016*, pages 1–6. IEEE Computer Society, 2016.
- [35] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, pages 1–6. IEEE, 2016.
- [36] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, volume 9906 of *Lecture Notes in Computer Science*, pages 869–884. Springer, 2016.
- [37] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance. *IEEE Trans. Multim.*, 20(3):645–658, 2018.
- [38] Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Lingyu Duan. VERI-Wild: A Large Dataset and a New Method for Vehicle Re-Identification in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3235–3243. Computer Vision Foundation / IEEE, 2019.
- [39] Elena Luna, Juan C SanMiguel, Jose M Martínez, and Marcos Escudero-Viñolo. Online Clustering-based Multi-Camera Vehicle Tracking in Scenarios with overlapping FOVs. *arXiv preprint arXiv:2102.04091*, 2021.
- [40] Javier Marín, David Vázquez, David Gerónimo, and Antonio M. López. Learning appearance in virtual scenarios for pedestrian detection. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 137–144. IEEE Computer Society, 2010.
- [41] 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.
- [42] Hyeonjoon Moon and P Jonathon Phillips. Computational and performance aspects of PCA-based face-recognition algorithms. *Perception*, 30(3):303–321, 2001.
- [43] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, and Siwei Lyu. The 2019 AI City Challenge. In *CVPRW*, page 452–460, June 2019.
- [44] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Alice Li, Shangru Li, and Rama Chellappa. The 6th AI City Challenge. *CoRR*, abs/2204.10380, 2022.
- [45] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E. Lopez, Anuj Sharma, Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. The 5th AI City Challenge. In *CVPRW*, June 2021.
- [46] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th AI City Challenge. In *CVPRW*, page 2665–2674, June 2020.
- [47] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018.
- [48] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *CVPRW*, pages 588–589, 2020.
- [49] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, volume 9906 of *Lecture Notes in Computer Science*, pages 102–118. Springer, 2016.
- [50] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016.

- [51] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016.
- [52] Ergys Ristani and Carlo Tomasi. Features for Multi-Target Multi-Camera Tracking and Re-Identification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6036–6046. Computer Vision Foundation / IEEE Computer Society, 2018.
- [53] Germán Ros, Laura Sellart, Joanna Materzynska, David Vázquez, and Antonio M. López. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3234–3243. IEEE Computer Society, 2016.
- [54] Andreas Specker, Lucas Florin, Mickael Cormier, and Jürgen Beyerer. Improving multi-target multi-camera tracking by track refinement and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3199–3209, 2022.
- [55] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Etinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2443–2451. Computer Vision Foundation / IEEE, 2020.
- [56] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *CVPR*, pages 8797–8806, 2019.
- [57] Duong Nguyen-Ngoc Tran, Long Hoang Pham, Hyung-Joon Jeon, Huy-Hung Nguyen, Hyung-Min Jeon, Tai Huu-Phuong Tran, and Jae Wook Jeon. A robust traffic-aware city-scale multi-camera vehicle tracking of vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3150–3159, 2022.
- [58] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4627–4635. IEEE Computer Society, 2017.
- [59] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTs: Multi-Object Tracking and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7942–7951. Computer Vision Foundation / IEEE, 2019.
- [60] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTs: Multi-Object Tracking and Segmentation. *arXiv:1902.03604[cs]*, 2019. arXiv: 1902.03604.
- [61] Hai Wang, Xin Yuan, Yingfeng Cai, Long Chen, and Yicheng Li. V2I-CARLA: A novel dataset and a method for vehicle re-Identification based V2I environment. *IEEE Transactions on Instrumentation and Measurement*, 2022.
- [62] Mark Weber, Jun Xie, Maxwell D. Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, Aljosa Osep, Laura Leal-Taixé, and Liang-Chieh Chen. STEP: Segmenting and Tracking Every Pixel. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.
- [63] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*, pages 10359–10366. IEEE, 2020.
- [64] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017.
- [65] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *CVPR*, pages 4256–4265, 2016.
- [66] Xipeng Yang, Jin Ye, Jincheng Lu, Chenting Gong, Minyue Jiang, Xiangru Lin, Wei Zhang, Xiao Tan, Yingying Li, Xiaoqing Ye, et al. Box-grained reranking matching for multi-camera multi-target tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3096–3106, 2022.
- [67] Hui Yao, Zhizhao Duan, Zhen Xie, Jingbo Chen, Xi Wu, Duo Xu, and Yutao Gao. City-scale multi-camera vehicle tracking based on space-time-appearance features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3310–3318, 2022.
- [68] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating Content Consistent Vehicle Datasets with Attribute Descent. In *ECCV*, page 775–791, August 2020.
- [69] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Based 3D Object Detection and Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11784–11793. Computer Vision Foundation / IEEE, 2021.
- [70] Zhedong Zheng, Tao Ruan, Yunchao Wei, and Yezhou Yang. VehicleNet: Learning Robust Feature Representation for Vehicle Re-identification. In *CVPRW*, volume 2, page 3, 2019.
- [71] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking Objects as Points. *ECCV*, 2020.