applied sciences

MDPI

Article

Investigating a Toolchain from Trajectory Recording to Resimulation

Florian Lüttner [1,*], Malte Kracht [2], Corinna Köpke [1], Annette Schmitt [2,3], Mirjam Fehling-Kaschek [1], Alexander Stolz [1,4] and Alexander Reiterer [2,3]

[1] Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI, 79588 Efringen-Kirchen, Germany; corinna.koepke@emi.fraunhofer.de (C.K.); mirjam.fehling-kaschek@emi.fraunhofer.de (M.F.-K.); alexander.stolz@emi.fraunhofer.de (A.S.)

[2] Fraunhofer Institute for Physical Measurement Techniques IPM, 79110 Freiburg, Germany; malte.kracht@gmail.com (M.K.); annette.schmitt@ipm.fraunhofer.de (A.S.); alexander.reiterer@ipm.fraunhofer.de (A.R.)

[3] Department of Sustainable Systems Engineering (INATECH)—Monitoring of Large Infrastructure, Albert Ludwig University of Freiburg, 79110 Freiburg, Germany

[4] Department of Sustainable Systems Engineering (INATECH)—Resilience of Technical Systems, Albert Ludwig University of Freiburg, 79110 Freiburg, Germany

* Correspondence: florian.luettner@emi.fraunhofer.de

Abstract: The growing variety of transportation options and increasing traffic congestion pose new challenges for road safety. As a result, there is an intensified focus on developing automated driving features and assistance systems aimed at minimizing accidents caused by human errors. The creation of these systems requires a substantial amount of testing kilometers, with estimates suggesting that around 2.1 billion kilometers would be necessary to ensure that each situation pertinent to the driving function is encountered at least once with a probability of 50%. This paper advances the microscopic simulation of traffic scenarios beyond linear patterns, utilizing the open-source environment openPASS. It addresses the research question of whether existing microscopic simulations are able to realistically represent non-linear traffic scenarios. A comprehensive toolchain integrates simulation with video recordings and laser scans. The study compares recorded traffic flow data with simulations at a T-junction, assessing the realism of vehicle models and trajectory representation. Three scenarios are analyzed, considering vehicles and pedestrians. The 3D geometry of the scene was captured with a laser scanner, enabling the mapping of recorded video data onto a geo-referenced environment. Object trajectories were extracted using an 'Regions with Convolutional Neural Networks features' object detector. While openPASS simulated vehicle and pedestrian behaviors effectively, limitations in trajectory variability and reaction times were observed. These findings highlight the need for more realistic behavior models. This research emphasizes the necessity for improvements to accommodate complex driving behaviors and pedestrian dynamics.

Keywords: traffic simulation; machine learning-based trajectory extraction; vehicle-pedestrian interaction; behavioral models

check for updates

Citation: Lüttner, F.; Kracht, M.; Köpke, C.; Schmitt, A.; Fehling-Kaschek, M.; Stolz, A.; Reiterer, A. Investigating a Toolchain from Trajectory Recording to Resimulation. *Appl. Sci.* **2024**, *14*, 10682. https://doi.org/10.3390/app142210682

Academic Editor: Matteo Prussi

Received: 30 September 2024
Revised: 11 November 2024
Accepted: 14 November 2024
Published: 19 November 2024

1. Introduction

An increasingly diverse range of transportation options and increasing traffic densities present new challenges for road traffic safety. Consequently, there is a growing effort to implement new automated driving features and assistance systems, decreasing the likelihood of accidents due to human errors [1]. The development of these systems requires a significant number of test kilometers for training and validation. In this context, the kilometers required for testing an automated driving function were estimated. To ensure that every situation relevant to the driving function is encountered at least once with a 50% probability, approximately 2.1 billion kilometers would be needed [2]. One opportunity to minimize the required test kilometers is realistic microscopic traffic simulations. These traffic flow simulations are used to represent the behavior of recorded road traffic participants via driver types and behavioral models.

There exist several traffic flow simulation environments that enable the representation and prognosis of traffic scenarios. In this context, the simulation environment PTV Vissim should be mentioned, which is, e.g., used to simulate traffic on intersections considering traffic lights and pedestrians [3]. Aimsun is another simulation environment [4] that primarily represents accident-free traffic. A brief overview of traffic flow simulation environments can be found in [5]. Furthermore, the simulation environment SUMO (Simulation of Urban MObility) [6] is a microscopic open-source traffic flow simulation without accidents. Specific behavior models are implemented in SUMO, such as an Artificial Neural Network (ANN)-based vehicle gap acceptance decision approach [7]. SUMO can be employed to analyze road networks that can be, e.g., generated via perception algorithms like YOLO5 (you only look once) [8]. OpenPASS is based on SUMO and optimized to represent critical scenarios via behavioral models (see, e.g., [9]). Thus, it is suitable to virtually analyze automated driving assistance systems (ADAS) [10]. Behavioral models can also be based on so-called stochastic cognitive models (SCM), which have been applied to highway traffic [5]. The developments in the openPASS simulation environment are being driven forward by Original Equipment Manufacturer (OEMs) in the automotive industry, such as Mercedes Benz AG, VW, and BMW, but also by Tier 1, such as Robert Bosch AG, and are partly incorporated into the simulation environment made available as open source.

Solutions have been presented in the past to simulate highway car traffic on linear roads and validate behavior models based on data (see, e.g., [11,12]). More complex scenarios, such as inner-city car traffic involving street crossings, have been studied by [13]. They simulate vehicle traffic in a specific area and account for the effects of traffic lights; however, Vulnerable Road Users (VRUs) are not included in the analysis.

Validation of the presented simulation environments is a crucial step for reliable prediction. This is typically conducted via video recordings where trajectories are extracted for vehicles and other road users. Recorded trajectories can be extracted with the help of Machine Learning (ML)-based motion tracking methods. An overview of existing motion tracking methods is given in [14–16]. Instead of optical sensors, e.g., Light Detection and Ranging (LiDAR) sensors can be employed to record traffic scenarios in transportation infrastructure [17]. This was demonstrated to identify pedestrians that aim to cross a road (see e.g., [18,19]). LiDAR can also be employed in combination with optical sensors to record the street scene [20].

The comparison of recorded and simulated data and trajectories is commonly employed to validate microscopic simulations. In existing literature, Automated Driving Functions (ADF) are virtually implemented for crash scenarios and compared with real-world data (see e.g., [10]). A comparison of macroscopic properties of Radar-recorded and simulated trajectories within SUMO is presented in [21]. A similar approach is followed by [22] for an inner-city scenario and statistical simulations with SUMO. A comparison of SUMO and Aimsun is performed via statistical simulations for linear traffic in [23]. Resimulation of real-world scenarios based on screenplay without considering behavioral models is presented in [24].

This paper seeks to assess the realism of behavioral models in traffic flow simulation environments. To exemplify this investigation, the open-source simulation framework openPASS and its associated behavioral models are employed to compare real-world trajectories with simulated ones from the same scenario. Due to the ability to depict critical scenarios and accidents in the simulation environment, the open source nature of the software, and the involvement of leading OEMs and Tier 1 suppliers in the development, the openPASS environment was chosen to realize the simulations.

Therefore, the setup of a semi-automatic toolchain is implemented from the recording of traffic scenarios and the classification using ML-based methods to the re-simulation using the agent-based, microscopic traffic flow simulation openPASS [25] and its behavioral models. The recorded and simulated trajectories are compared, also considering VRUs. The term scenario is used synonymously with concrete scenarios, according to Bagschik et al. [26]. In the work presented in this paper, we study traffic situations at a T-junction in Freiburg, Germany, involving cars and VRUs.

The remainder of this paper is organized as follows: First, in Section 2, the T-junction scenario is described; the recorded data, the extraction of trajectories, and the traffic simulation setup are presented. In Section 3, the simulation results are compared with the recorded data. Further, in Section 4, the results are discussed with respect to limitations in the simulation approach. Finally, Section 5 summarizes the findings and provides a brief outlook.

2. Material and Methods

The goals of this work are threefold. First, we aim to record scenarios in non-linear environments, where linear traffic is, e.g., represented by highway traffic. Second, we seek to extract trajectories using ML techniques. Finally, we will investigate the reproducibility of these traffic scenarios using the openPASS simulation environment, focusing on the underlying behavioral models (see Figure 1).

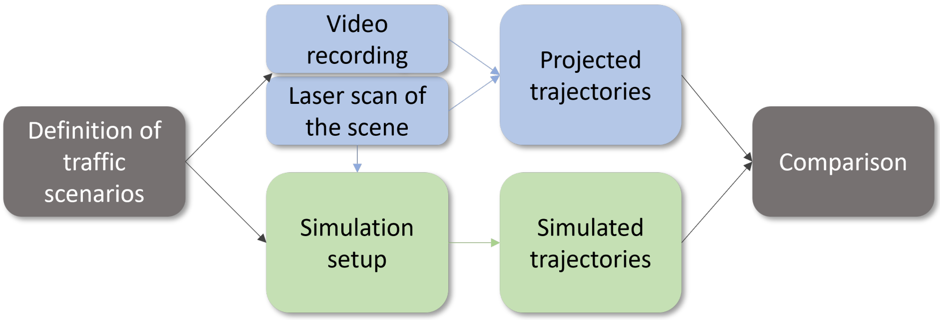


Figure 1. Flowchart of the toolchain presented in this paper.

For recording such scenarios, a T-junction located next to the Fraunhofer IPM building was selected. Due to data protection reasons, we staged the scenarios with volunteers. The advantages of this junction are that we could set up the camera on top of the Fraunhofer IPM, and due to the low frequency of traffic at the intersection, we were able to stage the scenarios.

For simulating scenarios in openPASS, it is necessary to choose parameters that are defined in the behavioral models that the simulation is using. Additionally, the scenery (T-junction) has to be created in the Association for Standardization of Automation and Measuring Systems (ASAM) OpenDrive standard [27] and the scenario, which defines the initial positions and routes of the simulated agents in the ASAM OpenScenario standard [28].

For modeling the T-junction in OpenDrive, the geometry of the junction is needed. Therefore, we use a terrestrial laser scanner, the Leica RTC360 [29], to record the 3D geometry of the T-junction. Using a laser scanner has the advantage that the 3D geometry of the scenery is captured very detailed in less than half an hour. From the point cloud, the kerbs and the color markings are extracted. This serves as the basis for the modeling. With the extracted geometric properties of the T-junction, the OpenDrive file can be created.

2.1. Traffic Scenarios

In total, we defined three different scenarios, which can be divided into two categories based on the road users involved, i.e., (i) cars to cars:

1. Two cars drive one behind the other. The car in front brakes until its velocity is zero ($v = 0$ km/h, m/s). The car behind reacts to this maneuver.

2. One car drives straight, while another car turns left and crosses the path of the first one, a so-called left-turn-across-path (LTAP) maneuver.

and (ii) cars to pedestrians:

3. Pedestrians are crossing the street in front of a car, which responds by braking to ensure their safety.

The three scenarios are presented visually in Figure 2. Car drivers were following a screenplay, whereas pedestrians were crossing the street randomly following traffic rules. Each scenario has been recorded once and resimulated also once.

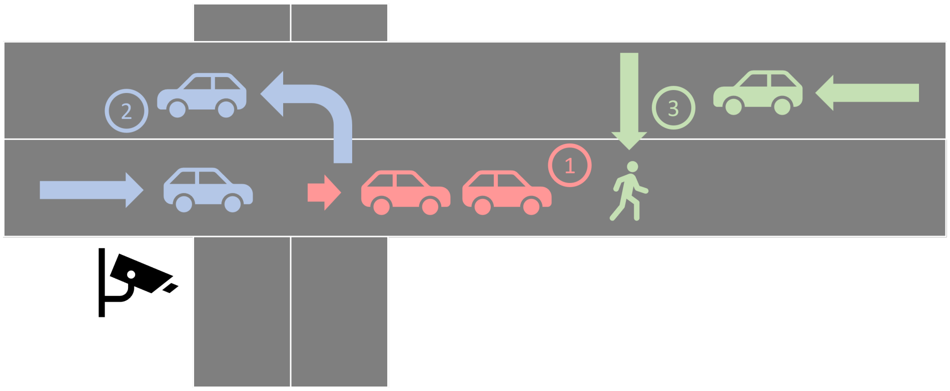


Figure 2. The three scenarios depicted on the T-junction. A car interferes with another road user, namely (1, red) a breaking car in front, (2, blue) a car in the LTAP maneuver, and (3, green) a crossing pedestrian. The camera position is also presented.

2.2. Recordings and Trajectory Extraction

To record the movement of the traffic participants in the scenery, a full-frame Sony Alpha 7R (Sony Corporation, Bangkok, Thailand) camera with a fixed focal length lens (Sony FE35F1.8) was mounted on the roof of Fraunhofer IPM. From the rooftop, it was not possible to film the scenery in a top view; therefore, a laser scan of the T-junction was made. We filmed the different scenarios with a frame rate of 100 fps and an image size of 1920 $\times$ 1080 pixels to detect the trajectories. With the help of a laser scan and the homography, the images were projected onto the street plane.

In order to overlay the camera recordings with a 2D top-down image of the laser scan recordings, the video output has to be distorted. With OpenCV, this can be reached by computing the homography matrix for an image and a planar surface or a reference image of the scene in orthographic projection [30]. Points that display the same real-world position can be marked with their respective pixel coordinates and will feed the computation of the matrix. At least four fixed points are needed to successfully compute the matrix, but more points increase the accuracy of the final image projection. Therefore, 55 points are used for the transformation.

For the object detection, we chose a Regions with Convolutional Neural Networks features (R-CNN) object detector based on the MobileNetV2 [31] CNN, which is pre-trained on a 1000-class ImageNet dataset [32]. MobileNetV2 is a very efficient and versatile network architecture and can easily be fine-tuned to a specific use case. Since our model targets a very specific scenario with a fixed camera perspective and only one type of car to detect, it was fine-tuned using additional training data for cars and pedestrians obtained from video images.

The system needs to be calibrated once to adjust for boundary conditions, such as cars driving into the street scene. As only the center point of the detected objects is used to extrapolate the trajectories, a car that is not fully visible yet will output a shifted center point, which would be interpreted as a change in velocity.

To correct the shifted data points, we use the static character of the system. First, we simply cut out the first few points where the center point of a viewed car is not actually visible. Afterward, we can calculate how far the center point is off from the ground truth by sampling video data. Since the car length and camera angle do not change, the error is the same in every video. We can simply calculate a common correction factor and adjust the data points.

After the object detection, trajectories for every car and pedestrian within the scenario can be extracted by tracking the objects in the projected street plane. Several objects, e.g., trees and street signs, within the scene can disturb this tracking. To adjust for these influences, we used the median between the detected road user center points of ten consecutive frames, so we ended up with 10 points per second. The final resulting trajectories are used to derive the speeds of the investigated traffic participants.

2.3. Traffic Simulation

For the re-simulation of scenarios, the open-source simulation environment openPASS (open platform for the assessment of safety systems) is used. The platform has been under constant development since 2016, also with the support of the automobile industry and corresponding suppliers.

The simulation platform openPASS can be used for stochastic traffic simulation by choosing statistical parameters of the simulated static agents or for the simulation of specific scenarios with dynamic agents. The behavioral models on which the simulation environment is based are the vehicle following model according to Krauß [9] and the lane change model according to Kraizewicz [33]. The corresponding equations of both models can be found in [34]. These behavioral models were originally designed to represent accident-free traffic. Accordingly, there have to be developments that enable the simulation to generate critical situations, including accidents. With respect to the behavioral models included in openPASS, the simulation platform is able to simulate linear traffic like on highways or interurban roads. The main actions on these types of roads are following other vehicles or changing the lane to overtake other vehicles. For simulating non-linear scenarios like urban intersections, it is expected that the simulation platform needs additional behavioral models for non-linear maneuvers like LTAP or crossing pedestrians. For setting up the simulation, nine config files are needed, including an ASAM OpenDrive file for defining the scenery and an OpenScenario file for defining the positions and routes of the simulated agents. In addition, catalog files are needed for defining the vehicles and the driver properties. Especially the driver properties influence the driving behavior of the simulated agents according to the underlying behavior models. In a slaveConfig file, the simulation is finally set up, and environmental parameters like weather conditions, visibility distances, and friction parameters are set, as well as parameters for spawning agents in the simulation and the name and directory of the simulation output file, although the environmental parameters have no influence on the simulation results for now.

ASAM OpenDrive is an open-source, standardized, xml-based language for defining properties of roads and whole road courses. These files can be input files for simulation platforms like openPASS to set up the simulated scenery. As described in the introduction, the geometry of the re-simulated T-junction was used to write such an OpenDrive file. For the simulation environment openPASS, OpenDrive version 1.0 was used.

In addition to the OpenDrive standard, ASAM OpenScenario is another open-source, standardized, xml-based language for defining a scenario. For use in openPASS, the OpenScenario file defines the simulated agents, the starting points of them, and the route. All interactions between the simulated agents should result from the behavioral models of the simulation interface openPASS. Without a simulation interface, the OpenScenario file can also be used for defining whole trajectories and storylines of actions and maneuvers at different time steps, such as breaking, lane changes, and overtaking. Due to the preliminaries of the used simulation environment, openPASS, the standard version 1.0 is used.

The behavioral models in openPASS (Krauß and Krajzewicz) are especially the models that are defined in the SUMO simulation platform [6]. In addition, the behavioral models were extended by new functions and parameters for generating critical situations and accidents. Such parameters are the cooperative of vehicles for lane changing when a faster vehicle is following, the wish of keeping on the right lane when approximating a slower vehicle, the accepted minimum distance to the vehicle in front dependent on the current velocities of both, and the willingness to violate the speed limit.

The behavioral model of Krauß describes an accident-free, location-continuous, and time-discrete modeling of following vehicles. The wish velocity $v_{safe}(t)$ of the following vehicle is therefore dependent on the velocity of the leading vehicle as a function of time ($v_l(t)$), the distance to the vehicle ahead as a function of time ($g(t)$), the reaction time of the driver (τ), the mean velocity of the following vehicle ($\overline{v}$), and a breaking term ($b(\overline{v})$). To generate error-prone driving, a randomly generated value ϵ_{error} is subtracted from the accident-free chosen wish velocity $v_{safe}(t)$. The maximum value for the deceleration of the vehicle ahead is estimated by the following vehicle. If the deceleration is greater than the estimated one, the own deceleration can exceed the wish deceleration. The equations for calculating the safety velocity of the following vehicle at each time are

$$v_{safe}(t) = v_l(t) + \frac{g(t) - v_l(t)}{\frac{\overline{v}}{b(\overline{v})} + \tau} \tag{1}$$

$$v_{wish}(t) = v_{safe}(t) - \epsilon_{error} \tag{2}$$

The behavioral model of Krajzewicz describes lane changing with tactical and strategic properties. Strategic lane changes are necessary lane changes for maintaining the route chosen. Tactical lane changes can be done to increase the velocity. The decision for a tactical lane change is made in two steps. The first step is to check if the distance to the next linking point of the route is big enough for not missing it while making a lane change. The second step is to check if the advantage (A) of driving in the neighboring lane would be sufficient. This advantage is calculated by the difference between the possible velocity on the neighboring lane (v_{nb}) and the own velocity (ego velocity v_{ego}) on the current lane divided by the maximum ego velocity ($max(v_{ego})$). If this term is positive, there is an advantage to changing the lane.

$$A = \frac{v_{nb} - v_{ego}}{max(v_{ego})} \tag{3}$$

The threshold for an advantage that is sufficient can be chosen. If the term is negative, a lane change would be a disadvantage. Note that in the scenarios considered in this paper, this behavioral model does not play a major role.

The parameters that were chosen to simulate the described scenarios with dynamic agents are parameters for setting up the route of the simulated agents (start positions, roads, lanes, velocities on lane) in the OpenScenario, parameters for defining the drivers properties and so the type of driver in the profiles catalog, parameters for defining the vehicles properties in the vehicle catalog, and rudimentary parameters for defining pedestrians in the pedestrians catalog. The parameters for sight distance and weather conditions in the slave config file are set to a default value. The parameters to choose are given in Tables 1–3.

In the vehicles catalog, many more parameters are defined, such as the gear ratio of the engine or the maximum torque, but it turned out that the simulation results are not sensitive to these parameters.

In the profiles catalog, any number of driver profiles and vehicle profiles on the basis of the vehicles catalog can be defined. The combination of driver and vehicle is also defined in the profiles catalog as agent. Each agent has a unique name that is used for identification in the OpenScenario file. The chosen parameters for the profile catalog are shown in Table 2.

Table 1. Parameters in the vehicle catalog.

Parameter	Scenario 1		Scenario 2		Scenario 3	
Name	car 1	car 2	car 1	car 2	car	ped
Vehicle category	car	car	car	car	car	pedestrian
Center of mass x [m]	1.49	1.49	1.49	1.49	1.49	0.20
Center of mass y [m]	0.00	0.00	0.00	0.00	0.00	0.00
Center of mass z [m]	0.75	0.75	0.75	0.75	0.75	0.90
Dimensions x [m]	2.16	2.16	2.16	2.16	2.16	0.40
Dimensions y [m]	5.27	5.27	5.27	5.27	5.27	0.30
Dimensions z [m]	1.49	1.49	1.49	1.49	1.49	1.80
Max. speed [m/s]	69.44	69.44	69.44	69.44	69.44	1.38
Max. deceleration [m/s^2]	9.81	9.81	9.81	9.81	9.81	9.81

Table 2. Parameters in the profile catalog.

Parameter	Scenario 1		Scenario 2		Scenario 3	
Agent mame	Ag1	Ag2	Ag1	Ag2	Ag1	Ag2
Type (D = dynamic/S = static)	D	D	D	D	D	D
Driver profile	Reg1	Reg2	Reg1	Reg2	Reg	Ped
Vehicle profile (TFV = TFVehicle)	TFV	TFV	TFV	TFV	TFV	TFV
Velocity wish [m/s]	11	14	12.5	6.0	10.0	1.38
Delta in velocity wish [m/s]	10	30	1.0	1.0	1.0	0.0
T gap wish to vehicle in front [s]	0.0	0.0	0.0	0.0	0.0	0.0
Min. distance [m]	14	0.1	10.0	10.0	10.0	10.0
Max. acceleration [m/s^2]	1.8	1.7	1.8	1.8	1.8	4.8
Max. deceleration [m/s^2]	9.81	9.81	9.81	9.81	9.81	9.81
Speed gain per time step [m/s]	0.0	0.0	0.0	0.0	0.0	0.0
Keep right	0.1	0.1	0.1	0.1	0.1	0.1
Cooperative (lane change)	0.4	0.4	0.4	0.4	0.4	0.4

Table 3. Parameters in OpenScenario.

Parameter	Scenario 1		Scenario 2		Scenario 3	
Name	car 1	car 2	car 1	car 2	car	ped
Scenario object	Ag1	Ag2	Ag1	Ag2	Ag1	Ag2
Road ID	3	3	3	2	3	7
Lane ID	−2	−2	−2	−2	−2	−1
Agent starting point [m]	108.0	77.0	135.0	65.0	25.0	3.8
Absolute target speed [m/s]	10.0	22.0	6.0	9.0	0.0	3.0
Road position s [m]	182.0	170.0	131.0	1.0	32.0	17.0
Road position t [m]	−2.0	−2.0	−2.0	−2.0	−2.0	−1.0
Simulation time condition [s]	80		80		80	

The parameters in Table 3 have to be chosen for each simulated agent. In addition, it is possible to define stories for each agent to give them predefined trajectories or maneuvers, such as a lane change for some time or distance-dependent conditions in the simulation. Due to this feature of the OpenScenario standard, it is possible to recreate trajectories of vehicles virtually in a script by using OpenDrive and OpenScenario files alone, but without simulated interaction between the virtual vehicles. OpenPASS simulates these interactions via the included behavioral models. Condition-dependent maneuvers according to a script can therefore only be represented to a limited extent in the openPASS simulation environment, since the simulation itself controls the behavior of the agents.

For each considered scenario, there are two protagonists to extract and re-simulate. The goal is to check if the behavioral models that are included in openPASS are able to re-simulate the recorded trajectories at a T-junction. Therefore, one vehicle is defined as the actor and the other as the reactor. Based on the given trajectory of the actor, the behavioral models of the reactor should lead to a trajectory that matches the recorded counterpart of the reactor. Therefore, we compare the trajectories and the velocities of the recorded and simulated vehicles over time.

3. Results

3.1. Recorded Data

First, the scenarios were recorded with optical sensors. The camera position resulted in a view as shown on the left side of Figure 3. From the rooftop, it was not possible to film the scenery in a top view. Therefore, a laser scan of the T-junction was made; see Figure 3 in the middle. The image transformed by homography is shown on the right side of Figure 3.

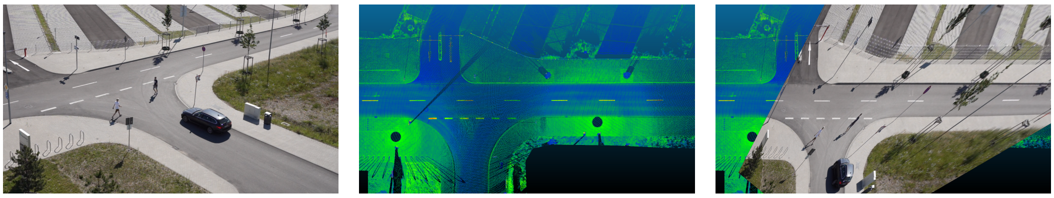


Figure 3. The input image from the perspective of the recording camera (**left**), the laser scan image (**middle**), and the resulting merged view obtained through transformation (**right**) are shown.

Next, trajectories of road users are extracted from the recorded videos. Figure 4 shows the raw data points of a trajectory, here exemplified for scenario 2. The corresponding corrected trajectories are presented in Figure 5. Finally, Figure 6 shows the traveled distance of a car between x = 0 m and x = 60 m with a speed of around 35 km/h (9.72 m/s).

With increasing traveled distance of the car, the distance between the recording system and the car is also increasing. For smaller distances between the recording system and the car, the derived speeds seem to be quite accurate compared with the real speed, with variations between 34 km/h (9.44 m/s) and 36 km/h (10 m/s). With increasing distance between the recording system and the car, the variation of the derived speeds is increasing up to a range between 36 km/h (10 m/s) and 43 km/h (11.94 m/s). In addition to the increasing variation in the derived speeds for higher distances between the recording system and the investigated traffic participant, the derived total speed is increasing, too. It is especially difficult to estimate the speed of vehicles that are far away from the camera, as the distance each pixel represents in the 2D projection becomes larger. A complete trajectory is shown in Figure 5, which is computed from data as shown in Figure 4. The results of the trajectory derivation are manually checked for plausibility in the videos. For this purpose, all detected objects are projected into the frames of the videos. As these are initial investigations, no ground truth has been created yet.

In the following, the simulation results are presented next to the recorded trajectories for the three scenarios under consideration.

Figure 4. Original image of the T-junction with raw data of three different trajectories shown as blue, green, and red lines.

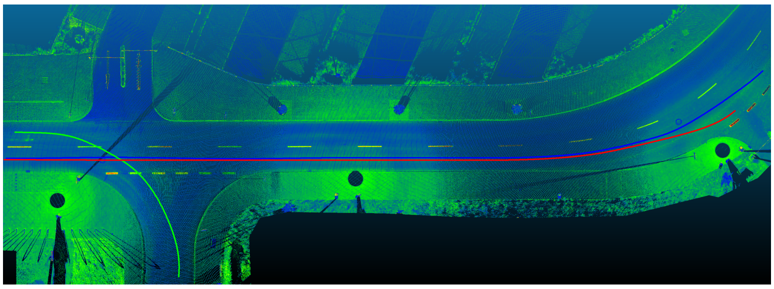


Figure 5. Final trajectories derived from adjustments shown as blue, green, and red lines with the laser scan of the intersection as background.

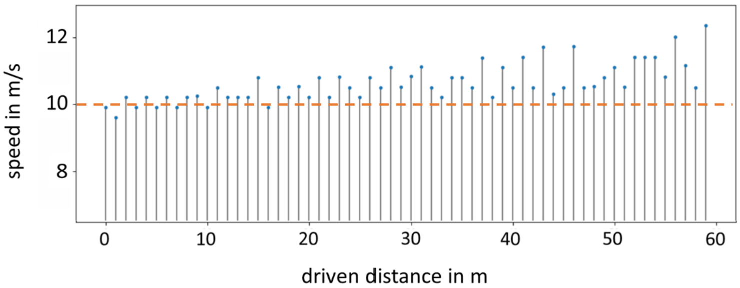


Figure 6. Visualization of the derived speed of a car from the recordings, depending on the distance traveled by the car. The actual velocity of the car, marked as a dotted orange line, is 35 km/h (9.72 m/s). The investigation of the car begins at x = 0 m and ends at x = 60 m. Consequently, the distance between the recording system (camera) and the investigated car also increases as the distance driven increases.

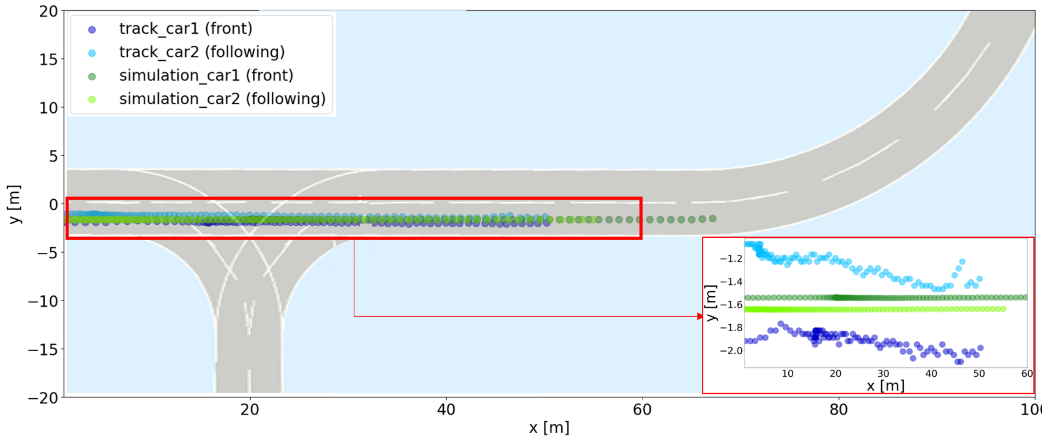
3.2. Scenario 1

In scenario 1, the actor can be accurately mapped for both the trajectories of the simulated and recorded vehicles, as well as their velocities. This forms the basis for examining the behavioral models contained in openPASS for their ability to reproduce the driving behavior of the reactor.

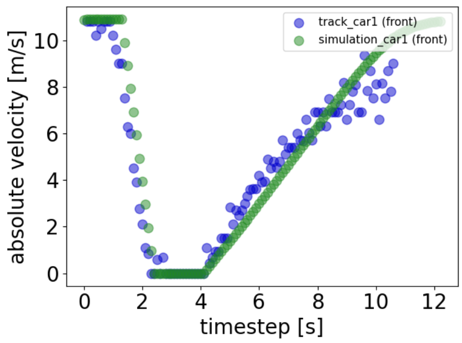
In Figure 7, slight deviations in the y-component of the recorded curves between 0.1 m and 0.5 m can be observed relative to the smooth curves of the simulation. These are due to two causes. The first one is based on uncertainties in the ML-based extraction of the trajectories from the video data. The second one is due to natural variations in steering by the driver. The simulated agents follow an idealized route in the middle of the lane without any of these variations. A realistic mapping of trajectories within the same lane is therefore limited by this idealization.

The velocities of the simulated and recorded vehicles show that driving behavior similar to the recorded trajectory can be achieved by the simulation. However, unlike the recorded vehicle, the simulated vehicle never comes to a complete stop. In addition, a temporal offset at the beginning of the curves can be observed. One contributing factor to the observed deviations is the reaction time, which is not appropriately set in the simulation software. The classical behavioral model, according to Krauß, considers a parameter for the reaction time as shown in Equation (1). Real values for reaction times are about 1 s to 2 s [35]. This corresponds approximately to the offset in time between the simulated and recorded curves. Further, the behavioral model without reaction time can avoid a collision at any time without coming to a stop for the distances in the scenario. This behavior is also justified by the low selected minimum distance of the reactor. However, larger minimum distances lead to even larger offsets of the curves in time in the simulation, which result from the lack of a realistic reaction time.

(**a**) Vehicle trajectories.



(**b**) Velocity of the leading vehicle, car 1.



(**c**) Velocity of the following vehicle, car 2.

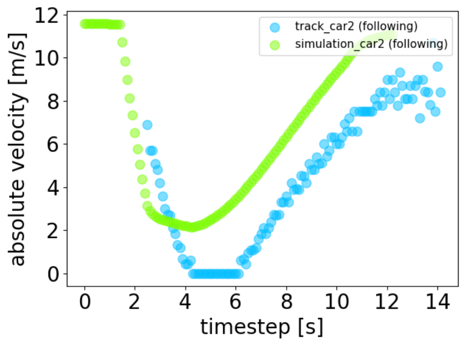


Figure 7. Results for Scenario 1: (**a**) Recorded (blue) versus simulated (green) vehicle trajectories; (**b**) Recorded (blue) versus simulated (green) velocities of the leading vehicle; (**c**) Recorded (blue) versus simulated (green) velocities of the following vehicle.

3.3. Scenario 2

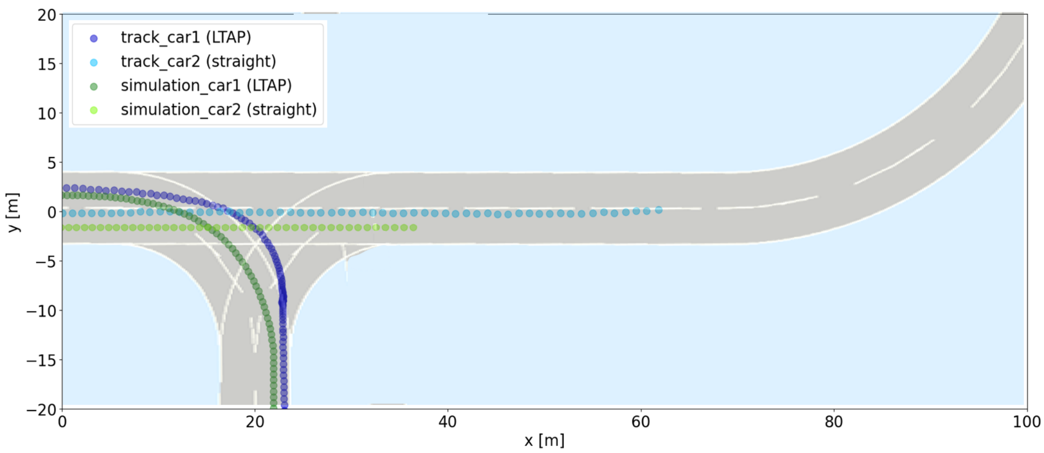
In scenario 2, two cars are passing each other, one going straight and one performing the LTAP maneuver. The corresponding recorded data are presented in detail in Figures 4 and 5. Note that for the comparison only one of the cars going straight has been considered.

Scenario 2 describes the LTAP maneuver. The trajectories in Figure 8a of the simulation and recording match well. For car 2, both the recorded and simulated trajectories are in the middle of the lane. For the left-turning vehicle car 1, the simulation again shows an idealized curve in the middle of the lane, while for the recorded vehicle, slight deviations of 0.1 m to 1 m can be recognized.

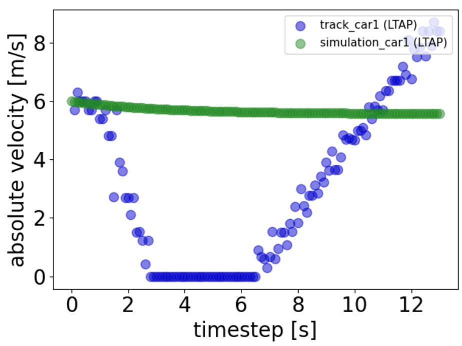
However, the progressions of the velocities present obvious deviations. Especially for car 1 in Figure 8b, a clear difference between simulation and recording can be observed. While in the recording the left-turning vehicle brakes and comes to a stop to let car 2 pass, the simulated car 1 hardly reduces its speed and performs its maneuver without the influence of car 2. It is clear that the behavioral models for lane change and vehicle following included in the simulation are not sufficient to represent such a scenario, as they only consider vehicles immediately next to or in front of them. However, the version of the simulation used does not yet include any functions to represent the observed behavior.

Nevertheless, such maneuvers are a common cause of accidents because of the intersecting trajectories. In Figure 8c, it can be seen that the simulated car 2 reduces its speed and reacts to car 1 as it enters the intersection to avoid a collision. This behavior is realistic, analogous to scenario 1, except for the offset of about 1 s due to the lack of reaction time. The implemented behavioral models seem to be suitable for the targeted simulation of critical scenarios (accidents and near-accidents) and the resulting braking behavior. However, evasive maneuvers are not considered. The large scatter in the velocity for the recorded car 2 is again due to the ML-based tracking of the trajectories.

(**a**) Vehicle trajectories.



(**b**) Velocity of the crossing vehicle, car 1.



(**c**) Velocity of the vehicle going straight, car 2.

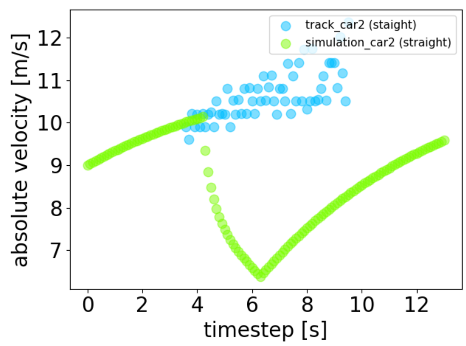


Figure 8. Results for Scenario 2: (**a**) recorded (blue) versus simulated (green) vehicle trajectories; (**b**) recorded (blue) versus simulated (green) velocities of the crossing vehicle; (**c**) recorded (blue) versus simulated (green) velocities of the vehicle traveling straight.

3.4. Scenario 3

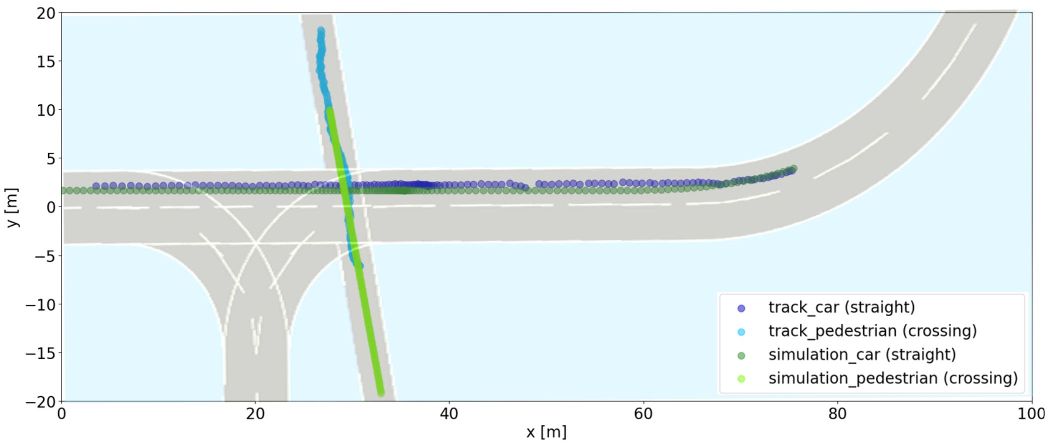
In scenario 3, we study an encounter between a car and a pedestrian. The velocity of the pedestrian increases while crossing the street, which is also reflected in the simulation.

Figure 9 shows the trajectories of scenario 3. The used simulation environment does not include pedestrian behavioral models. Thus, the pedestrian is identified as the actor in scenario 3. The trajectory of the pedestrian could be reproduced well. As in the previous scenarios, the trajectory of the reactor car 1 shows slight deviations from the simulated trajectory, which are due to the idealized driving in the simulation in the middle of the lane and slight fluctuations of the trajectory from the tracking. Nevertheless, the trajectories can be reproduced well.

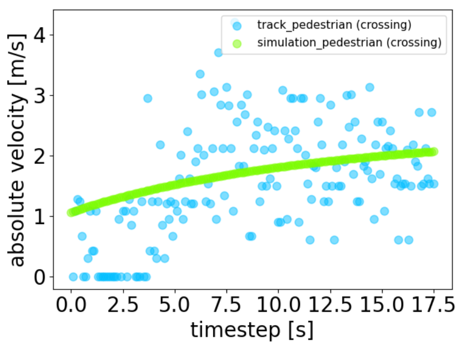
Due to the small cross-section of pedestrians in the videos, the tracking of these exhibits variation in the direction of movement and is orthogonal to it. This leads to extreme fluctuations in the calculation of the absolute velocity in each time step between 0 and 4.2 m/s, which can be seen in Figure 9b. As mentioned above, the derived velocity varies, so the velocity is smoothed using the median over 10 data points. Figure 9c shows the velocity profile of the vehicle in scenario 3. As in scenario 1, the time gap of about 1 s between the onset of braking in the simulation and the recorded trajectory can be observed in scenario 3. The lack of reaction time of the simulated agents leads to the earlier braking maneuver. The acceleration of the simulated car 1 decreases after stopping until reaching the desired velocity, while it seems to be linear in the real course. However, both courses are realistic.

Additionally, the simulated car 1 exhibits a shorter duration of standstill compared with the recorded vehicle. The behavioral models do not consider a lateral safety distance. Thus, car 1 does not wait until the pedestrian has crossed the road. As soon as the pedestrian has left the cross section of car 1, it accelerates again. Except for the limitations described above, the velocity profile of the recorded vehicle can be reproduced well.

(**a**) Vehicle and pedestrian trajectories.



(**b**) Velocity of the pedestrian.



(**c**) Velocity of the vehicle.

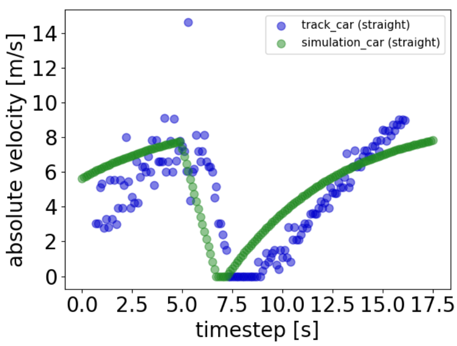


Figure 9. Results for Scenario 3: (**a**) Recorded (blue) versus simulated (green) vehicle trajectories; (**b**) Recorded (blue) versus simulated (green) velocities of the pedestrian; (**c**) Recorded (blue) versus simulated (green) velocities of the vehicle.

3.5. Quantitative Comparison

A quantitative comparison of velocities between recorded and simulated trajectories was performed using the Root Mean Square Error (RMSE) presented in Table 4. The spatial deviation of trajectories has been defined by the RMSE of the Distance of the Nearest Neighbors (DNN) between locations along the trajectories.

These values underline the visual interpretation of the comparison between recorded and simulated trajectories for each scenario. Significant deviations are most prominent for the non-linear scenarios, which are due to systematic errors as described above and the existing behavioral models that are not suitable for non-linear scenarios. The large deviations in velocity and spatial distance in scenario 2 result from the lack of traffic rules in the simulation. For scenario 3, the deviation stems from the simulated car that does not consider a lateral safety distance, i.e., to wait until the pedestrian has left the street entirely before continuing to drive.

Table 4. RMSE of velocities for the three scenarios.

Scenario		RMSE [m/s]	RMSE DNN [m]
1	Car 1 (front)	1.046	0.381
	Car 2 (following)	3.048	0.470
2	Car 1 (LTAP)	3.791	2.049
	Car 2 (straight)	3.189	9.373
3	Car (straight)	2.183	0.564
	Pedestrian (crossing)	1.378	0.269

4. Discussion

There are two major limitations that have been mentioned in Section 3, i.e., the estimation of velocities from the recorded data and the behavioral models within the traffic simulation. These are discussed in more detail in this section.

The data capture using a camera have to be optimized with respect to the derivation of the velocity, especially for pedestrians. Due to the angle of exposure and the transformation into a plane, the pedestrians in the pictures are distorted. That makes it difficult to calculate the correct center point for the trajectory. Therefore, it is recommended to optimize the calculation of the pedestrian center points. The calculation of the car's center points also needs improvement. Additionally, the methods for smoothing the velocity estimates need to be refined.

In the version of the traffic simulator openPASS used in this research, pedestrians are only implemented very simply and without a behavior model. They follow predefined trajectories without reacting to the surrounding traffic. Thus, only the behavior of the vehicles can be examined for a replication of the recorded trajectories as they react to the pedestrians movement.

Further, an interesting traffic scenario could not be considered for this research because of limitations in the simulator, i.e., overtaking maneuvers in the proximity of the T-junction with traffic on both lanes. The issue is that the OpenDrive standard only specifies a single direction of travel for each lane. The modeled T-junction describes traffic from all directions towards and away from the intersection, according to their lanes. Thus, it is not possible to model an overtaking maneuver and traffic on both lanes at the same time considering defined driving directions on the lanes relative to the defined reference line in the OpenDrive standard. The reference line takes the geometry of the road, while the lanes are defined next to this line. The driving direction on the lanes is defined by a sign and a number. In the direction of the reference line, the lanes are numbered ascending and negative (-1, -2, $-n$), while the lanes are correspondingly ascending positive (1, 2, m). For this reason, a workaround for the used OpenDrive standard is proposed.

The idea is to define overlaying roads with different driving directions and no connectors on the crossing as they are defined in the standard. For all possible routes on the t-crossing from the start to the end of the route, a road with the corresponding reference line and two lanes according to the driving direction on the route is defined; see Figure 10. This realization defines an OpenDrive-based file that includes all possible driving directions on the T-junction, and respective openPASS simulations are able to interpret the scenario.

It should be mentioned that for larger scenarios with more than one intersection and a more complex routing, this kind of workaround may not be practical. Furthermore, more research and development is needed for the proper interpretation of the respective OpenScenario files within openPASS.

However, for investigating single scenarios on intersections or parts of roads, it seems to be a suitable way to enable a more realistic application of the open-source simulation platform openPASS. It offers the possibility to define an intersection scenery and, at the same time, simulate traffic from any possible direction on arbitrary lanes in openPASS. Although the defined agents drive on roads with different IDs, they recognize each other and react in the way defined by the behavioral models.

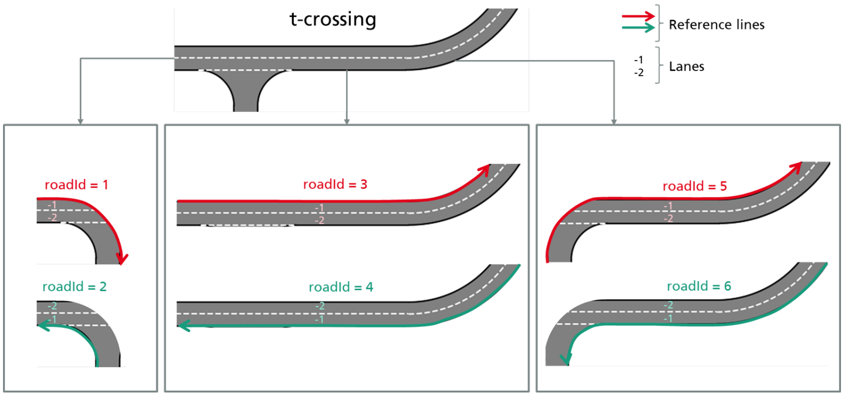


Figure 10. Alternative OpenDrive implementation of the investigated intersection to enable the mapping of additional interesting maneuvers in the scenery, independent of the predefined direction of travel established by the OpenDrive standard. The examined T-junction was divided into three sections (grey boxes), each further subdivided into two directions of travel (red and green reference lines). By overlapping the resulting six sections, it becomes possible to represent scenarios such as overtaking maneuvers with traffic on both lanes.

5. Conclusions

This paper contributes to the efforts of simulating traffic scenarios that go beyond linear traffic, e.g., on highways. An open-source simulation environment is employed, namely openPass. Furthermore, the paper presents a toolchain that integrates the simulation and the comparison to video recordings and laser scans.

In this study, a comparison between recorded and simulated traffic flow data was presented. Initially, scenarios on a T-junction involving cars and VRUs demonstrated that the simulation models for vehicles are quite realistic. The scenery, as well as the trajectories recorded in it, could be depicted realistically. Three scenarios were studied, i.e., (1) two cars driving behind each other, (2) one car drives straight and the other turns left, and (3) a pedestrian is crossing the street in front of a car.

The 3D geometry of the scenery was captured with a laser scanner, and the recorded video data of the three scenarios could be mapped via transformation on the geo-referenced environment. The extraction of vehicle and pedestrian trajectories was performed using the R-CNN object detector, which was able to successfully track all objects. However, the algorithm had difficulties in estimating the velocity of objects. Further research and a comparison of object detection algorithms are needed. Additionally, a detailed evaluation of the object detection is necessary. Therefore, a geo-referenced data set of vehicle and pedestrian trajectory should be created, e.g., by total station-based object tracking.

The open-source simulation environment openPASS was used to simulate the three scenarios. It was able to represent the behavior of cars and pedestrians. However, the idealized type of trajectory formation in the simulation does not allow any fluctuations of the trajectory within the lanes. This was observed as a deviation between recorded and simulated trajectories. However, this property had no significant influence on the course of the scenarios, so this effect can be neglected for the recorded scenarios. Nevertheless, a more realistic movement on the lane is needed for the investigation of other important maneuvers like avoiding obstacles in the lane, overtaking, or scenarios with traffic on both lanes. Additionally, the version of the simulation environment used lacks a realistic parameter for road users' reaction times, resulting in earlier vehicle reactions than what is observed in actual recorded trajectories. However, since reaction times can be assumed, the trajectories could be corrected accordingly afterwards. Turning maneuvers such as an LTAP can only be represented to a limited extent by the included behavioral models. Vehicles do not react to approaching opponents unless they are in front of them (vehicle following model) or beside them (lane change model). This results in critical scenarios (near misses and accidents) and enables an investigation of these critical scenarios resulting from missing reactions to approaching vehicles. For more complex, non-linear driving scenarios, the simulation environment needs further enhancements. Finally, behavioral models of pedestrians and other VRUs need to be incorporated in the simulation environment.

Author Contributions: Conceptualization, F.L., C.K., and A.S. (Annette Schmitt); methodology, F.L., C.K., and A.S. (Annette Schmitt); software, F.L. and M.K.; validation, F.L.; formal analysis, A.S. (Annette Schmitt); investigation, F.L., A.S. (Annette Schmitt) and C.K.; data curation, A.S. (Annette Schmitt), and M.K.; writing—original draft preparation, F.L., C.K., and A.S. (Annette Schmitt); writing—review and editing, C.K., M.F.-K., A.R., and A.S. (Alexander Stolz); visualization, F.L., A.S. (Annette Schmitt) and M.K.; supervision, A.S. (Alexander Stolz), M.F.-K. and A.R.; project administration, C.K. and M.F.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This work received funding from Sustainability Centre Freiburg through the project SURF-Multisafe and from the German Federal Ministry for Economic Affairs and Climate Action through the project KIsSME, funding number 19A20026B.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors acknowledge the use of FhGenie, a Fraunhofer-owned chat bot, to correct the grammar in this document. Moreover, the authors extend their thanks to the reviewers for their insightful comments, which played a crucial role in enhancing the quality of this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADAS	Automated Driving Assistance System
ADF	Automated driving function
ANN	Artificial Neural Network
ASAM	Association for Standardization of Automation and Measuring Systems
CNN	Convolutional Neural Network
DNN	Distance of Nearest Neighbors
LiDAR	Light Detection and Ranging
LTAP	Left-Turn-Across-Path
ML	Machine Learning
OEM	Original Equipment Manufacturer
R-CNN	Regions with Convolutional Neural Networks features
RMSE	Root Mean Square Error
SCM	Stochastic Cognitive Model
SUMO	Simulation of Urban MObility
VRU	Vulnerable Road User
YOLO	You only look once

References

1. Severino, A.; Curto, S.; Barberi, S.; Arena, F.; Pau, G. Autonomous Vehicles: An Analysis Both on Their Distinctiveness and the Potential Impact on Urban Transport Systems. *Appl. Sci.* **2021**, *11*, 3604. [CrossRef]
2. Wachenfeld, W.; Winner, H. Die Freigabe des autonomen Fahrens. In *Autonomes Fahren*; Maurer, M., Gerdes, J., Lenz, B., Winner, H., Eds.; Springer: Berlin, Germany, 2015; pp. 439–464.
3. Ziemska-Osuch, M.; Osuch, D. Modeling the assessment of intersections with traffic lights and the significance level of the number of pedestrians in microsimulation models based on the PTV Vissim tool. *Sustainability* **2022**, *14*, 8945. [CrossRef]
4. Aimsun. Aimsun Next 23 User's Manual, Aimsun Next 23.0.0, Barcelona, Spain. 2023. Available online: https://docs.aimsun.com/next/23.0.0/ (accessed on 13 November 2024).
5. Fries, A.; Fahrenkrog, F.; Donauer, K.; Mai, M.; Raisch, F. Driver behavior model for the safety assessment of automated driving. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), IEEE, Aachen, Germany, 5–9 June 2022; pp. 1669–1674.

6. Krajzewicz, D.; Hertkorn, G.; Rössel, C.; Wagner, P. SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In *A. Al-Akaidi (Hg.): Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002), Sharjah, United Arab Emirates, 28–30 September 2002*; SCS Publishing House: Erlangen-Brück, Germany, 2002; pp. 183–187.
7. Bagheri, M.; Bartin, B.; Ozbay, K. Simulation of vehicles' gap acceptance decision at unsignalized intersections using SUMO. *Procedia Comput. Sci.* **2022**, *201*, 321–329. [CrossRef]
8. Shirazi, M.S.; Morris, B.T.; Zhang, S. Intersection analysis using computer vision techniques with SUMO. *Intell. Transp. Infrastruct.* **2023**, *2*, liad003. [CrossRef]
9. Krauß, S. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics;* Report Number: DLR-FB-98-08, available from TIB Hannover RN 437(98-08), 1998; pp. 1–126. ISSN: 1434–8454. Available online: https://sumo.dlr.de/pdf/KraussDiss.pdf (accessed on 13 November 2024).
10. Fahrenkrog, F.; Hammouda, M.; Fischer, F.; Maier, L. Virtual Simulation Based Assessment of ADAS in Consumer Tests by openPASS. In Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV), National Highway Traffic Safety Administration, Yokohama, Japan, 3–6 April 2023.
11. Eisemann, L.; Fehling–Kaschek, M.; Gommel, H.; Hermann, D.; Klemp, M.; Lauer, M.; Lickert, B.; Luettner, F.; Moss, R.; Neis, N.; et al. An approach to systematic data acquisition and data-driven simulation for the safety testing of automated driving functions. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2003; pp. 2440–2447.
12. Eisemann, L.; Fehling-Kaschek, M.; Forkert, S.; Forster, A.; Gommel, H.; Guenther, S.; Hammer, S.; Hermann, D.; Klemp, M.; Lickert, B.; et al. A Joint Approach Towards Data-Driven Virtual Testing for Automated Driving: The AVEAS Project. In Proceedings of the Fast-Zero'23, Full Online Conference: Program & Proceedings, Society of Automotive Engineers of Japan, Inc. (JSAE), Online, 22–24 May 2024.
13. Bindzar, P.; Saderova, J.; Sofranko, M.; Kacmary, P.; Brodny, J.; Tutak, M. A Case Study: Simulation Traffic Model as a Tool to Assess One-Way vs. Two-Way Traffic on Urban Roads around the City Center. *Appl. Sci.* **2021**, *11*, 5018. [CrossRef]
14. Liu, G.; Liang, X.; Yu, H.; Lu, Z.; Kang, K.; Li, T.; Liu, B. Video Moving Object Detection Technology Based on Deep Learning. In Proceedings of the 2021 China Automation Congress (CAC), IEEE, Beijing, China, 22–24 October 2021; pp. 221–225.
15. Gururaj, V.; Varada, S. R.; Satheesh, S.; Kodipalli, A.; Thimmaraju, K. Analysis of deep learning frameworks for object detection in motion. *Int. J. Knowl.-Based Intell. Eng. Syst.* **2022**, *26*, 7–16. [CrossRef]
16. Barbu, T.; Bejinariu, S.-I.; Luca, R. Transfer Learning-based Framework for Automatic Vehicle Detection, Recognition and Tracking. In Proceedings of the 2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, Iasi, Romania, 27–28 June 2024; pp. 1–6.
17. Wang, J.; Zhang, S.; Guo, H.; Tian, Y.; Liu, S.; Du, C.; Wu, J. Stereoscopic monitoring of transportation infrastructure. *Autom. Constr.* **2024**, *164*, 105472. [CrossRef]
18. Zhao, J.; Xu, H.; Wu, J.; Zheng, Y.; Liu, H. Trajectory tracking and prediction of pedestrian's crossing intention using roadside LiDAR. *IET Intell. Transp. Syst.* **2019**, *13*, 789–795. [CrossRef]
19. Guan, F.; Xu, H.; Tian, Y. Evaluation of roadside LiDAR-based and vision-based multi-model all-traffic trajectory data. *Sensors* **2023**, *23*, 5377. [CrossRef] [PubMed]
20. Reiterer, A.; Wäschle, K.; Störk, D.; Leydecker, A.; Gitzen, N. Fully Automated Segmentation of 2D and 3D Mobile Mapping Data for Reliable Modeling of Surface Structures Using Deep Learning. *Remote Sens.* **2020**, *12*, 2530. [CrossRef]
21. Schrader, M.; Al Abdraboh, M.; Bittle, J. Comparing Measured Driver Behavior Distributions to Results from Car-Following Models using SUMO and Real-World Vehicle Trajectories from Radar: SUMO Default vs. Radar-Measured CF model Parameters. *SUMO Conf. Proc.* **2023**, *4*, 41–54. [CrossRef]
22. Lobo, S.C.; Neumeier, S.; Fernandez, E.M.G.; Facchi, C. InTAS–The Ingolstadt Traffic Scenario for SUMO. *arXiv* **2020**, arXiv:2011.11995. [CrossRef]
23. Ronaldo, A.; Ismail, T. Comparison of the Two Micro-Simulation Software Aimsun & Sumo for Highway Traffic Modelling. 2012 https://www.diva-portal.org/smash/get/diva2:555913/FULLTEXT01.pdf (accessed on 13 November 2024) .
24. Punzo, V.; Simonelli, F. Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transp. Res. Rec.* **2005**, *1934*, 53–63. [CrossRef]
25. Dobberstein, J.; The Eclipse Working Group openPASS. An open source approach to safety impact assessment via simulation. In Proceedings of the ESV 2017 Organizing Committee: 25th ESV Conference Proceedings, 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV), Detroit, MI, USA, 5–8 June 2017. Available online: https://www-esv.nhtsa.dot.gov/Proceedings/25/25ESV-000094.pdf (accessed on 13 November 2024).
26. Bagschik, G.; Menzel, T.; Reschka, A.; Maurer, M. Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In Proceedings of the Workshop Fahrerassistenzsysteme und Automatisiertes Fahren, Walting, Germany, 29–31 March 2017.

27. Dupuis, M.; Strobl, M.; Grezlikowski, H. OpenDRIVE 2010 and Beyond-Status and Future of the de facto Standard for the Description of Road Networks. In Proceedings of the Trends in Driving Simulation Design and Experiments, Proceedings of the Driving Simulation Conférence EUROPE: 2010 Arts et méRiers pars Tech, Institut national de recherche sur les transports et leur sécurité—INRETS, Bron (Rhône), Paris, France, 9–10 September 2010; pp. 231–242. Available online: https://www.ifsttar.fr/fileadmin/user_upload/editions/inrets/Actes/Actes_INRETS_A126.pdf#page=233 (accessed on 13 November 2024).
28. Amid, G. OpenSCENARIO Concept Project. Association for Standardization of Automation and Measuring Systems (ASAM). 2019. Available online: https://www.asam.net/project-detail/openscenario-concept-project/ (accessed on 13 November 2014).
29. LEICA Geosystems AG. *Leica RTC360 3D Reality Capture Solution*; LEICA Geosystems AG: St. Gallen, Switzerland, 2018.
30. Available online: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html (accessed on 13 November 2024).
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
32. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
33. Krajzewicz, D. Kombination von taktischen und strategischen Einflüssen in einer mikroskopischen Verkehrsflusssimulation. *Fahrermodellierung Wiss. Wirtsch. Berl. Fachtag. Fahrermodellierung* **2009**, *28*, 104–115.
34. Hoffmann, S. *Mikroskopische Modellierung und Bewertung von Verkehrssicherheitskritischen Situationen*; Technische Universität München: München, Germany, 2019.
35. Johansson, G.; Rumar, K. Drivers' brake reaction times. *Hum. Factors* **1971**, *13*, 23–27. [CrossRef]