III. MATERIALS and METHODS

1. Creating simulation scenario and test case

The first subsection deals with the used Software to create the test environment and the test scenario itself. CARLA is an open-source urban driving simulator for autonomous driving research and supports flexible sensor suites and full control of all static and dynamic actors and maps [2]. The flexible API and the ROS integration provides a lot of flexibility and the possibility to extract the ground truth data directly from the scenario. The work is based on the CARLA 0.9.8 release combined with ROS and python3 packages and a python3.5 founded API. The test case is derived from the at Euro NCAP used Car-to-Pedestrian Nearside Child 50 % (CPNC-50) test scenario from the Insurance Institute for Highway Safety (IIHS) test protocol [3, 4].

TABLE 1: Test conditions pedestrian autonomous emergency braking (P-AEB) [4]

\begin{table}[]

\begin{tabular}{|l|l|}

\hline

\textbf{Parameter} & \textbf{CPCN-50 Scenario Child} \\ \hline

Test vehicle speed & 40 km/h \\ \hline

Pedestrian target speed & 5 km/h \\ \hline

Target direction & Crossing from R-to-L \\ \hline

Target path & Perpendicular \\ \hline

Pedestrian dummy size & Child \\ \hline

Overlap & 50 % \\ \hline

\end{tabular}

\end{table}

The test procedure starts with launching the test scenario by first spawning the three vehicles and the pedestrian to their initial positions into the map. This state consists of an Audi TT (1) in front and an Audi e-tron (2) arranged behind it. The left edges of both cars are parked 0.2 m away from the right edge of the test lane. The longitudinal distance between the cars and between the front car and pedestrian is 1.0 m, each. At the beginning of the simulation, the child pedestrian is positioned 7.0 m laterally from the center of the ego-vehicle, which is centered in its lane 200 m behind the pedestrian and portrayed as an Audi e-tron.

Bild:Target\_Placement\_test\_scenario einfügen

Fig. 1: Target placement based on CPNC-50 test [4]

After a few seconds, the ego vehicle starts accelerating quickly to 40 km/h and the child starts moving with constant speed from right-to-left to cross the street. The pedestrian becomes visible for the ego vehicle after he passed the Audi TT (2). The ego vehicle immediately engages an emergency brake and comes to a standstill in front of the child. At this point the pedestrian is at the 50 % overlapping point. The child continues crossing the street and the scenario ends as soon as the child completely passed the ego-vehicle.

1. Creating objects list of Ground-Truth Data

The Ground Truth Data Objects List includes four message vectors for every spawned object – Classification, Dimension, Features and Geometric. These messages are used to classify the objects, send their geometrical dimensions, location, acceleration, angle and visible edges [5].

The Classification parameters indicate the type of the spawned objects and differentiate between vehicle, pedestrian and other types. In addition, the Features vector contains all visible and invisible edges of the objects. The evaluation of both messages is statically generated for this scenario referring to the bounding box data of every spawned object. The third message represents the length, width and height of the object. This Dimensions vector receives the information as well from the bounding box. Furthermore, the Geometric message is used to represent the coordinates, speed, yaw angle and acceleration of the objects relative to the ego vehicle.

The Ground Truth Data will be published via ROS in two different topics. Every topic includes a header with timestamp, object ID as well as an Objects List message. This Objects List message includes all four messages for the pedestrian and both parked cars in topic one and is referenced to the center of the objects. Topic two includes only the Objects List message with the Geometric data of the ego vehicle based on the camera position at the front middle of the car.

The test scenario offers two options for publishing different data in topic one.

* Publishing only Objects in the field of view of the camera (200 m and a total opening angle of 60 °)
* Publishing all spawned objects over the whole test period

[2] https://carla.org/, abgerufen am 08.06.2020

[3] euro-ncap-aeb-vru-test-protocol-v302.pdf von Maikol/Fabio

[4] test\_protocol\_pedestrian\_aeb.pdf von Maikol/Fabio

[5] Aeberhard Dissertation