PRISSMA Project
Plateforme de Recherche et d'Investissement pour la Sûreté
et la Sécurité de la Mobilité Autonome
04/2021 - 04/2024

[DELIVERABLE 3.5] CARRYING OUT THE FIRST EVALUATION TEST CAMPAIGN IN A CONTROLLED ENVIRONMENT AND PRODUCING TEST REPORTS.

Main authors : Rémi Régnier(LNE), Mohamed Boudali (LNE), Pierre DUTHON (CEREMA), Jean-Baptiste Horel (INRIA), Khushdeep Singh Mann (INRIA), Alessandro Renzaglia (INRIA), Christian Laugier (INRIA), Alain Piperno (UTAC), J. Bouret, Mickael Dessaint (UTAC), Céline Serbouh (UTAC)

Keywords: Proof of concept, test report, augmented reality, simulation, test bench, AI based vehicles homologation, repeatability tests, robustness tests, AI overfitting tests, anticipation tests, closed road tests

Abstract. This deliverable is intended to provide the test reports from the first test campaign, which was carried out during the first half of 2023. This first campaign provides feedback for the improvement and enhancement of our protocols in order to improve the second campaign, which will take place at the end of the project and will validate the work of WP3. During the first campaign, 3 POCs were carried out, one on the use of augmented reality as a means of testing on tracks between INRIA and Transpolis, another on tests in degraded conditions on benches between CEREMA and LNE and finally one carried out by UTAC on the generalisation of current tests to better include AI aspects.

Résumé. Ce livrable se propose de fournir les rapports de tests de la première campagne d'essais qui a été réalisé durant le premier semestre de 2023. cette première campagne permet d'avoir un retour d'expérience pour l'amélioration et l'enrichissement de nos protocoles afin d'améliorer la seconde campagne qui se déroulera en fin de projet et permettra de valider les travaux du WP3. Durant la première campagne 3 POCs ont été réalisé, un sur l'utilisation de la réalité augmentée comme moyen de test sur piste entre l'INRIA et Transpolis, un autre sur les tests en conditions dégradées sur banc entre le CEREMA et le LNE et enfin un dernier réalisé par l'UTAC sur la généralisation des tests actuelle pour mieux y inclure les aspects IA.

Contents

List of Figures

List of Tables

1 INRIA / Transpolis POC

1.1 Introduction

This section presents the first experimental campaign conducted by Inria at the Transpolis testing facility in the framework of the PRISSMA project in February 2023. The tests, representing one of the POC proposed by the project, had the main objective of showing the interest and potentiality of using a new augmented reality framework [4] as a tool to improve testing and validation of AI-based algorithms in controlled environments. The key aspect of this method is the design of a merge function allowing a real-time augmentation of LiDAR data with virtual elements (see Fig. 1 for an illustrative example). With this solution, we open new possibilities for testing. A testing site can very easily be populated with many and diverse virtual elements in order to create more complex test scenarios. Virtual pedestrians or cars are easier to operate and offer richer and more active behaviors (e.g. reacting to the ego-vehicle's motion). Furthermore, all elements of the test scenario that may induce a collision risk can be replaced by their virtual counterpart to secure the tests in the early stages of development or to test the system in critical situations. Virtual scenarios are also repeatable and this is a key feature to reproduce experiments. Our AR[1] testing implementation accurately represents the virtual scenes and guarantees a consistent fusion with the real world. So AR tests produce meaningful results that can be used to infer the behavior of the vehicle in the real world. Finally, as any element can be either real or virtual, AR testing offers a smooth transition from simulation to actual testing. For these reasons, the proposed AR framework can be a fundamental testing solution for the validation of advanced automotive software.

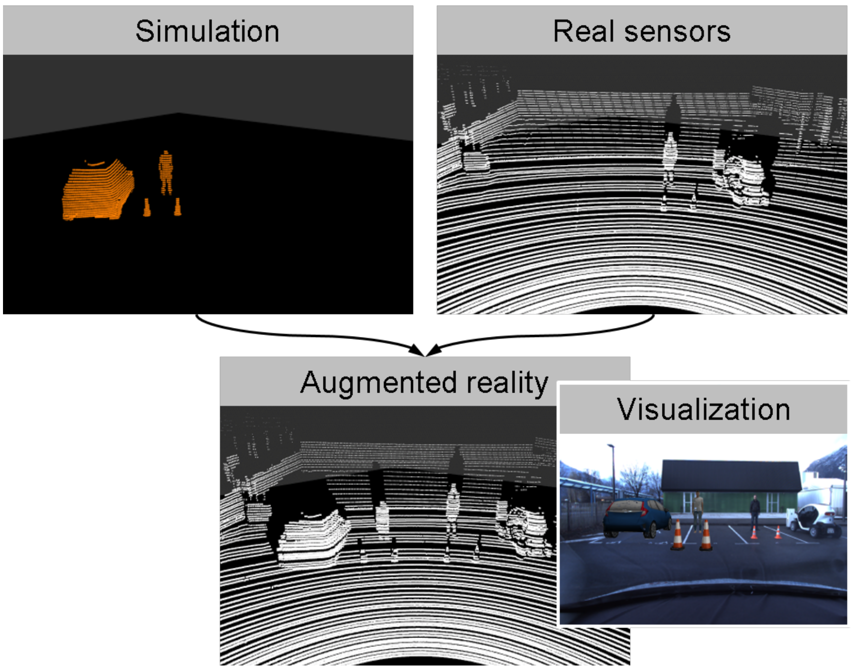


Figure 1: Example of LiDAR point cloud augmentation. The introduced virtual point cloud and initial sensor point cloud are shown on the top. The technique that we present, deployed on a vehicle, generated the fused point cloud and the visualization of the augmented scene on the bottom.

[1] AR: augmented reality

The tests have been carried out with the Inria's autonomous vehicle (Renault Zoé). A detailed description of the experimental platform and its sensors is given later. The main scope of the experiments was to study how the AI-based perception framework present on the vehicle reacts to the real-time fusion of real and virtual data obtained through the AR framework. in particular, the tested module is a Bayesian perception algorithm, the Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) [5]. The CMCDOT is a spatial occupancy tracker, which provides a dense and generic representation of the environment [6] through a probabilistic occupancy grid, based on Bayesian fusion, filtering of sensor data and Bayesian inference. It infers the dynamics of the scene and represents the environment with static and dynamic occupancy, free spaces and unknown regions. Furthermore, based on the occupancy and velocity estimations for each cell in the grid, a probabilistic collision estimation is also available. The navigation module utilizes then the information provided by the CMCDOT and its environment representation to make decision and, for instance, to activate emergency brake and obstacle avoidance manoeuvres.

The ego-vehicle has been tested in a series of safety-critical scenarios (also later described), all including virtual dynamic vehicles of different sizes (cars and trucks). in particular, some of them potentially led to a collision with the ego-vehicle that could be avoided only with an emergency brake maneuver. The AR framework fully proved its value in recreating in the Transpolis controlled area scenarios that would have been otherwise complex to reproduce and potentially dangerous.

The rest of this section gives more information on the experimental platform and modules used for this POC and a detailed description of the tests and the obtained results. As later explained, the current lack of an accurate ground truth for this first experimental campaign limited the possibility of obtaining a thorough quantitative analysis.

1.2 Augmented reality

Our AR system consists of the four following modules:

- a virtual environment which contains a twin of the experimental vehicle
- a synchronization module which updates the position and state of the virtual twin
- a sensor emulation which generates outputs from the virtual sensors and integrates them in the actual sensors' outputs
- a visualization which helps testers to understand the AR scene.

Fig. 2 proposes a schematic representation of the software framework. The periodic messages of the sensors of the real vehicle give rhythm to the virtual world. So all modules must run in real time, their execution duration must be short compared to the period of the sensors. This is a heavy constraint on the design and implementation of the solution.

We firstly generate a virtual environment which is anchored to a real world position with a reference in GPS coordinates. Then, the virtual environment contains only a virtual twin of the vehicle under test and the virtual elements that we want to add in augmented reality. There is no restriction on the virtual elements of the test scene. The scene can be as complex as required by the test and include any type of object, the only limits are the ones of the simulator. Apart of the virtual vehicle and the test elements, the virtual environment is empty. Our method does not need a background, a ground plane and any representation of the actual test site. This makes this method easy to deploy in a new place.

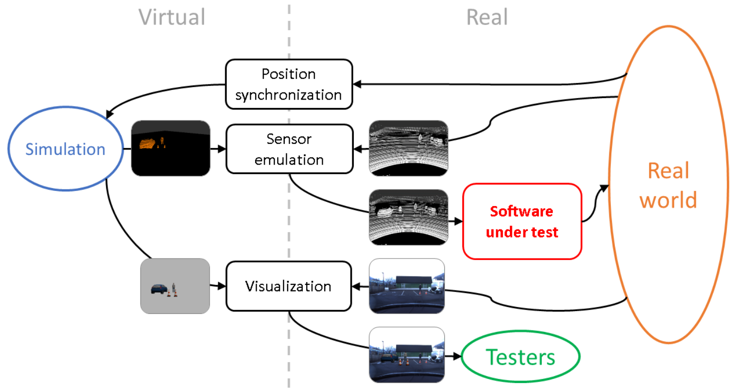


Figure 2: Structure of our AR framework.

The absolute position of the vehicle under test must be constantly estimated by an accurate localization system. The estimated position is used to set the position of the virtual twin of the vehicle under test in the virtual environment. This straightforward synchronization gives a great flexibility. The AR system can be deployed without any installation.

The virtual twin of the vehicle is equipped with a set of sensors that mimics the sensors of the actual vehicle. An accurate, realistic and real-time emulation of the sensors is needed. Although the framework is generic, for this POC we focus on LiDAR sensors. The emulated LiDARs must return the detection of the virtual objects under a point cloud format. The point clouds are then merged with those returned by each corresponding actual sensor. The merge process is a key component of the proposed AR framework: it must be real time despite the amount of data to process; it must consider a realistic sensor model; it must reproduce all occlusions between real and virtual world. For each sensor, the merge produces a new point cloud that represents the AR perception. It can then be sent to the software of the vehicle under test in place of the actual sensor's point cloud. Thanks to this, the use of AR is seamless for the software under test. For more technical details on how the LiDAR virtual and real data are merged in real time see [4].

The virtual twin of the vehicle is also equipped with a set of cameras that mimics the ones of the actual vehicle. Thanks to the simulator, the virtual cameras return images of the virtual objects. These images are then merged with those of each corresponding camera. For each camera, this produces a new image that represents the AR perception. It provides the testers with a convenient insight of the AR scene. If using a photo-realistic simulator and a realistic image merge function, this visualization can be used as AR for perception with cameras. However, a simulator with approximate graphics and a simple merge procedure suffice for the purpose of visualization.

1.3 Abstract scenario description

For this experiment, we defined one abstract scenario where the ego-vehicle crosses twice the same intersection. The Zoé is the only real actor of the scenario. The virtual actors are four: a fire truck, a bus, and two cars. The scenario is divided in two phases, Figure 3 shows the phase 1 of the scenario and Figure 4 shows the trajectories of the actors during the 2 phases.

1. The Zoé drives toward the intersection, simultaneously a bus and a car are coming from the right. The bus might reach the intersection at the same time as the Zoé: depending on the timing and experimental conditions it will lead to a collision or an avoidance or an

emergency brake. The car is occluded by the bus, thus not visible by the LiDARs, if the the Zoé avoids the bus without braking it might also collide with the car.

2. The Zoé drives toward the intersection (opposite lane of phase 1), simultaneously a fire truck and a car are coming from the right. The fire truck might reach the intersection at the same time as the Zoé, depending on the timing and experimental conditions it will lead to a collision or an avoidance or an emergency brake. If the Zoé performs a emergency brake in front of the fire truck, it might collide the car as it reaches the intersection after the fire truck.

The behavior of the Zoé during the scenario execution is hardly predictable. While the trajectories of the virtual obstacles are precisely controlled and deterministic, the navigation of the Zoé is non deterministic.

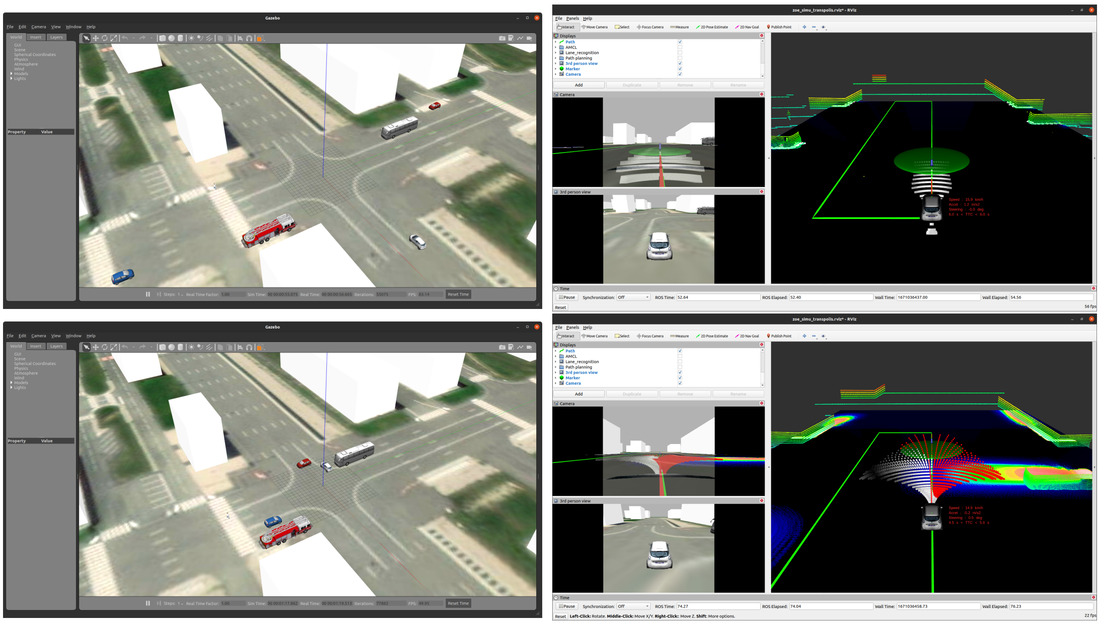


Figure 3: ROS RViz and Gazebo screenshots

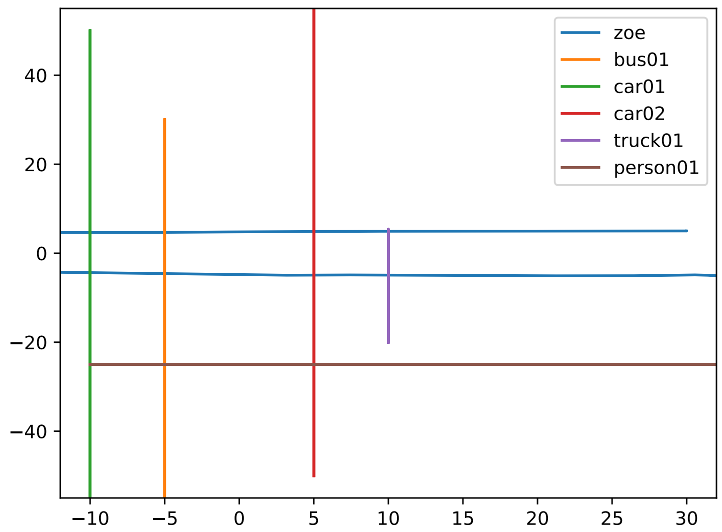


Figure 4: Actors trajectories

1.4 Perception module

The perception module in Fig. 5 relies on probabilistic occupancy grids and Bayesian fusion techniques to generate detailed and refined representations of occupancy and velocity. These representations can be effectively utilized in tasks such as planning, risk evaluation, and collision avoidance.

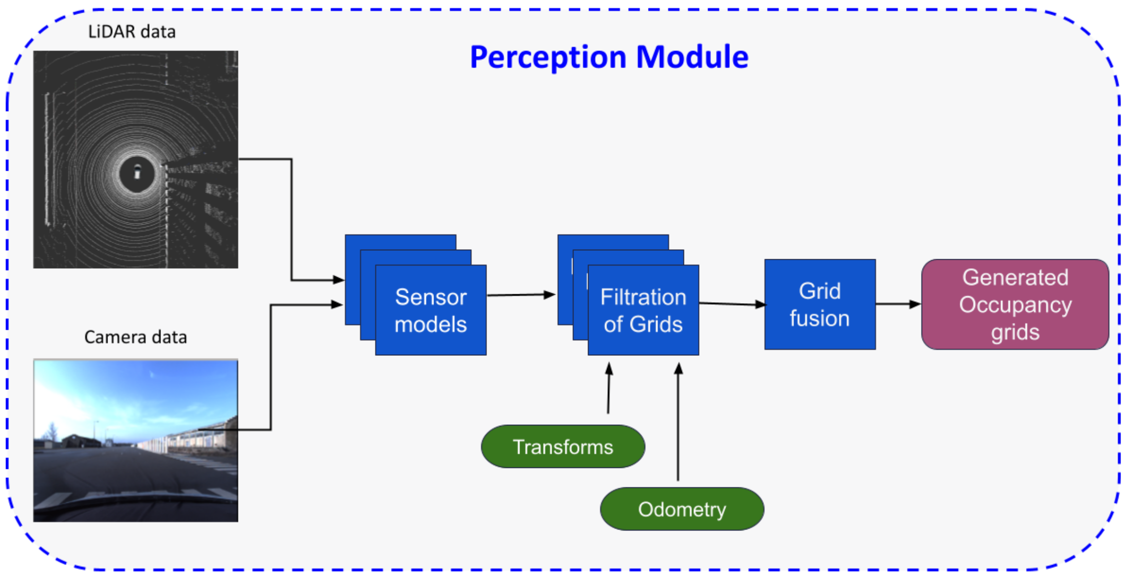


Figure 5: Overview of perception module: Generating detailed occupancy and velocity representations using probabilistic grids and Bayesian fusion for effective planning, risk evaluation, and collision avoidance.

In this POC, the perception module is represented by the CMCDOT framework, developed by Inria [5], which is a comprehensive method for tracking occupancy in dense environments. This approach draws inspiration from the Bayesian occupancy filter framework, incorporating abstract states and a conditional Monte Carlo technique to optimize velocity estimation and focus on relevant areas. The scene analysis encompasses static, dynamic, free, and unknown states, each associated with dedicated models. The method explicitly considers uncertainty and sensor coverage.

The CMCDOT modules takes the following inputs:

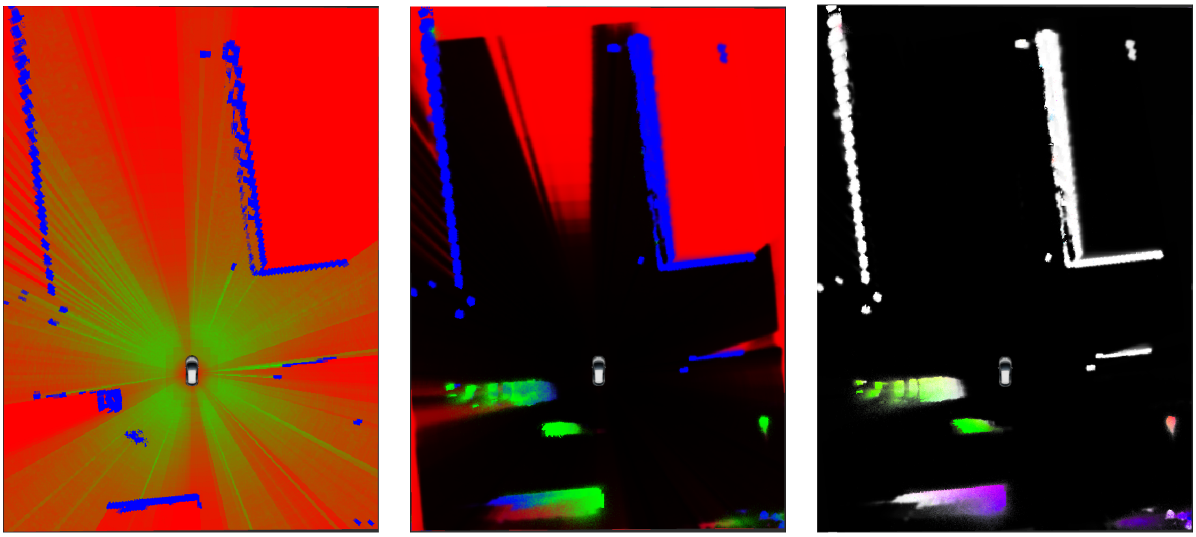
- LiDAR pointcloud data
- Observed occupancy grids or several grids from different sensors
- Odometry of the ego-vehicle

As a result, the CMCDOT module generates the following output grids visualized in Fig. 6:

1) **Instantaneous grid**: Initially, a Bayesian model is defined for each sensor. By considering a specific sensor measurement, the sensor model calculates the probabilities of occupancy in the 2D space surrounding the robot. This results in instantaneously updated occupancy grids.

2) **Filtered occupancy grid**: The instantaneous occupancy for each sensor modality is filtered in both time and space. The CMCDOT occupancy filter, utilizing a Bayesian update model, performs local occupancy filtering while also tracking occupancy changes using a Monte Carlo approach. This yields filtered occupancy grids and velocity grids.

3) **Velocity grid**: This grid visually represents stationary elements (shown in white) as well as dynamic obstacles (represented by various colors). The intensity of each color indicates the obstacle's speed, while the color itself signifies its direction of motion.



(a) Instantaneous grid: Unknown (red), static and dynamic occupancy (blue), free space (green)

(b) Filtered occupancy grid: Unknown (red), static occupancy (blue), dynamic occupancy (green), free space (black)

(c) Velocity grid: Static occupancy (white) and dynamic occupancy (varied color intensities)

Figure 6: Comprehensive Analysis of Occupancy and Motion: Instantaneous Grid (a), Filtered Occupancy Grid (b), and Velocity Grid (c)

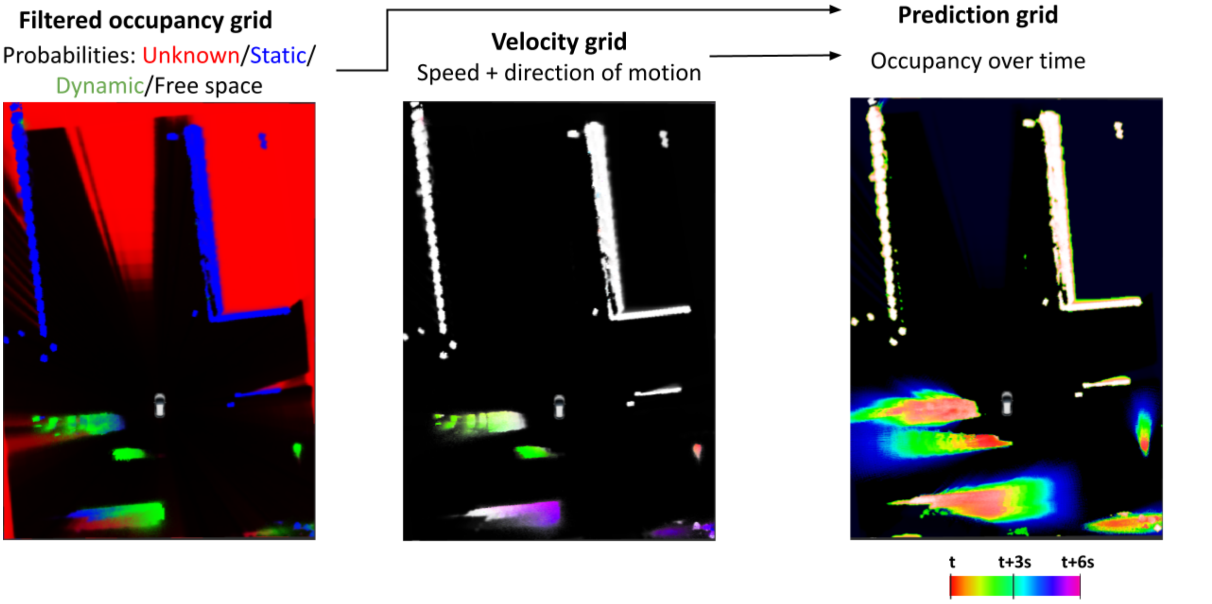


Figure 7: DWA grid: Predicts occupancy by projecting cells with velocity, incorporating noise, merging sensor data. Offers insights into future occupancies, integrates path planning and obstacle avoidance for comprehensive understanding of dynamic environments.

4) **DWA (Dynamic Window Approach) grid**: This grid is an effective model used for predicting occupancy by incorporating essential input data, such as occupancy probabilities and estimated velocity in Fig. (7). It projects each cell based on estimated velocity, enabling the representation of movement. To account for noise, cells are divided into particles with specific accelerations and angular velocities. The generation scheme for this grid as been proposed in [7]

Acting as a probabilistic distribution, the DWA grid provides insights into future occupancies within a three-second timeframe. It merges occupancy grids obtained from various sensor measurements, creating a unified representation that accumulates information over time. The velocity grid derived from lidar measurements is preserved as the most accurate estimation of motion.

In addition, the DWA prediction grid enhances occupancy understanding by visualizing predictions over time. Static objects are represented in white, while moving objects are depicted with colors based on their estimated time of arrival. To ensure conservative behavior near moving objects, a large uncertainty is introduced during the prediction process, resulting in the creation of clouds of predicted occupancy. This accounts for potential variations and uncertainties associated with object movement.

1.5 Navigation module

The navigation module in Fig. 8 plays a crucial role in guiding the vehicle through its environment. It encompasses several key steps to ensure safe and efficient navigation. Firstly, the module employs a dynamic window approach (DWA) by generating a list of feasible command samples, including acceleration, steering, and braking, for the next time window of 4 seconds.

These command samples are carefully selected through sampling commands software.

Next, the navigation module applies a kinodynamic model to compute the resulting trajectory for each command sample. It takes into account the vehicle's dynamics and constraints to determine the most suitable trajectory. Simultaneously, it leverages the occupancy and velocity grids to predict the occupancy over time, incorporating an uncertainty model.

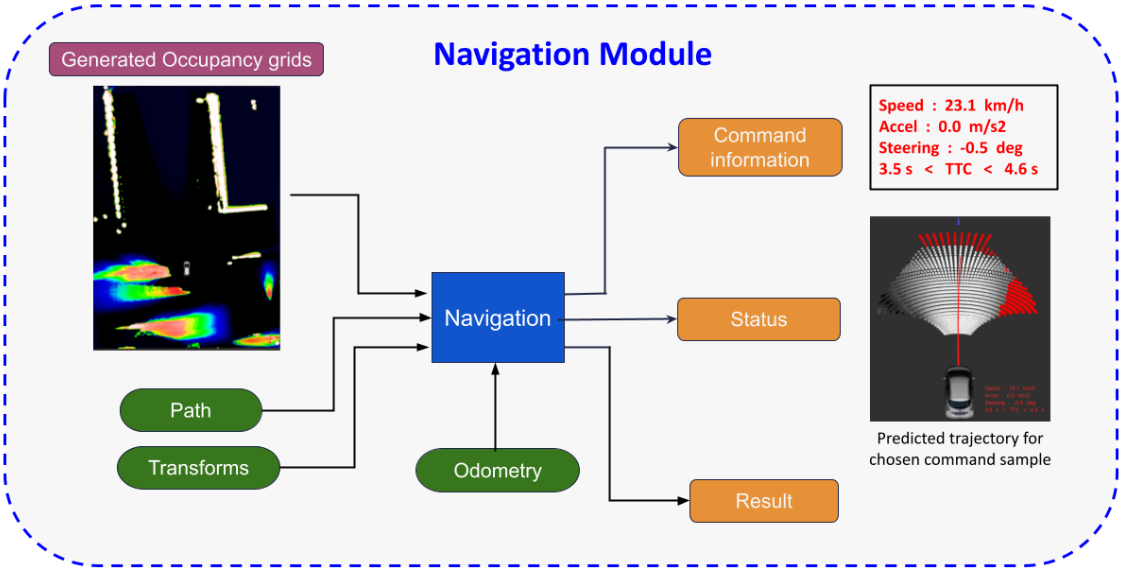


Figure 8: Overview of Navigation module: Generating feasible command samples, predicting occupancy, evaluating collision risk, and selecting the best command for optimal trajectory, ensuring safe and efficient navigation.

To evaluate collision risk, the navigation module computes the expected time to collision (TTC) for each command sample. Command samples that lead to unavoidable collisions are discarded to prioritize safety. The module then compares the remaining command samples using a cost function that combines collision avoidance and path following costs. This step aims to find the best command sample that balances both safety and optimal navigation. This type of collision detector serves as an interface between grid-based perception and sampling-based planners as described in [7].

Finally, the navigation module executes the selected command sample with the minimal cost. This command sample is forwarded to the controller, which translates it into vehicle actions, ensuring the vehicle follows the desired trajectory while effectively navigating the environment. Through this comprehensive process, the navigation module enables the vehicle to navigate efficiently, avoiding obstacles and minimizing collision risks.

1.5.1 Local Planner

The local planner encompasses several key components to ensure effective decision-making for the vehicle's trajectory. It starts by defining admissible values for acceleration and steering, generating a set of admissible commands. For each command, the planner predicts the trajectory and computes the Time to Collision (TTC). A cost function is then calculated for each command, leading to the selection of the best command.

The current trajectory refers to the predicted trajectory for the currently chosen command sample, typically for the next time window of approximately 2 seconds. Command information includes the predicted speed at the end of the current window, along with details and boundaries for predicting the TTC. When no path is set or when no command sample can be selected, no display is shown.

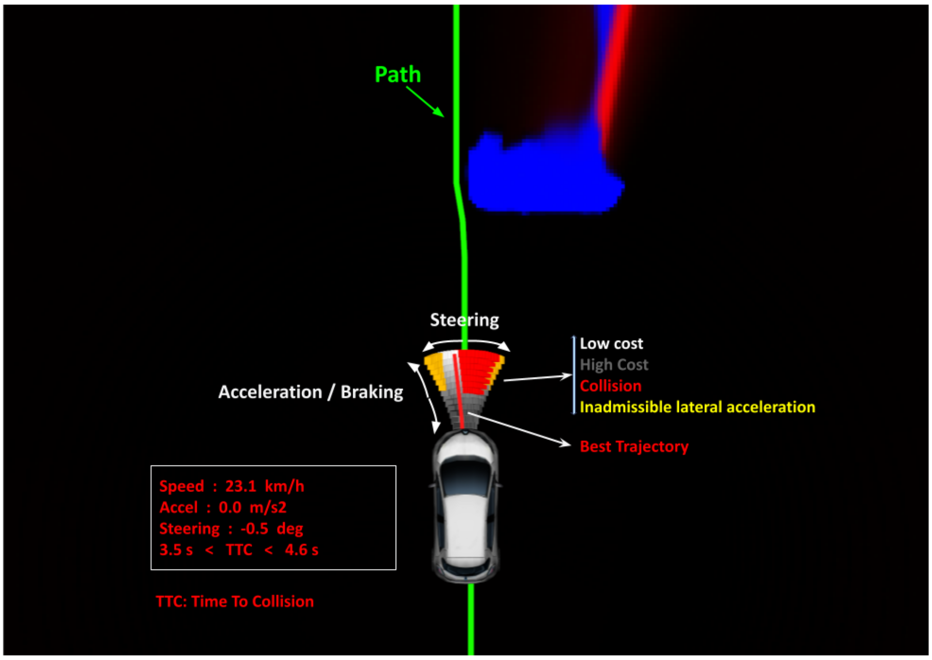


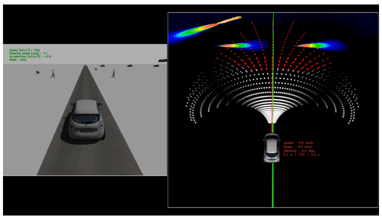
Figure 9: Illustration of Local Planner: Generating admissible commands, predicting trajectories, and evaluating collision risks. Accurate and efficient computations for optimal trajectory selection within short time horizons.

To evaluate different trajectories, the local planner predicts the positions at the end of the time window for all command samples that satisfy the speed and acceleration limits. Trajectories that result in a collision are represented by red squares, while trajectories with different cost values are represented by squares with a grayscale color (with white indicating the best).

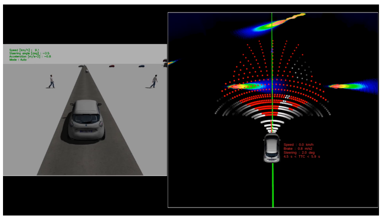
Accuracy is achieved through precise trajectory prediction, accurate representation of the ego vehicle shape, and predicting the motion of other agents. The local planner ensures computing efficiency through massively parallel computations over the ego vehicle positions and trajectories. Simplicity is maintained by focusing on simple trajectories and short-term predictions ranging from 5 to 10 seconds.

1.5.2 Local Planner: Illustration scenario stopping before a pedestrian

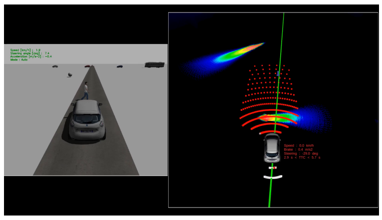
Example 1: Pedestrian crossing scenario, i.e, wherein pedestrians are predicted to obstruct the vehicle's path, a progressive deceleration strategy is employed to avoid collisions. For pedestrian safety, the vehicle comes to a complete stop, giving precedence to their presence on the path. Once pedestrians clear the path, motion resumes seamlessly to maintain an uninterrupted journey.



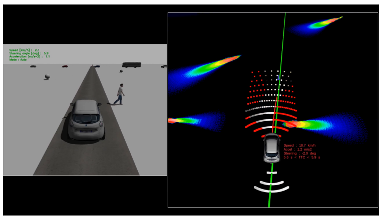
(a) Enhancing Safety: Slight deceleration implemented for pedestrians located far ahead.



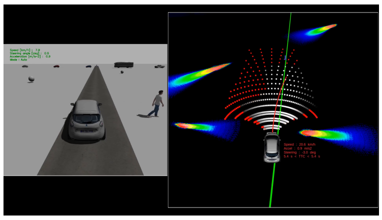
(b) Predictive Collision Avoidance: Progressive deceleration for obstructing pedestrians.



(c) Ensuring Safety: Complete stop to prioritize pedestrian presence on the path.



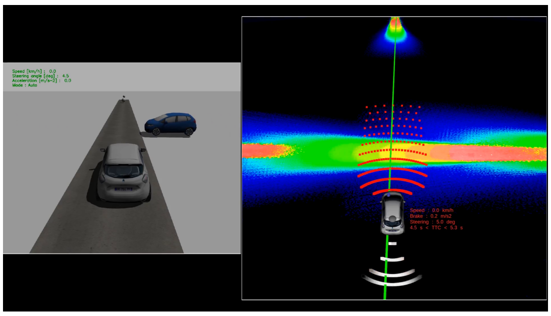
(d) Resuming Motion: Resumption of motion upon pedestrian path clearance, ensuring smooth travel.



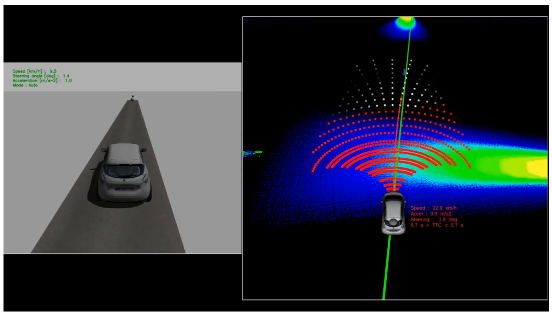
(e) Path Clarity: Acceleration engaged when no pedestrians expected, ensuring a clear trajectory.

Figure 10: Pedestrian crossing scenario: Optimizing safety through various measures, including deceleration for distant and obstructing pedestrians, complete stop for pedestrian presence, and acceleration for a clear and uninterrupted trajectory.

Example 2: Vehicle crossing scenario, i.e, as ego-vehicle encounters a line of fast-moving vehicles, it comes to a halt, prioritizing caution and safety. The ego vehicle then skillfully identifies a window of opportunity within the queue, allowing it to proceed and cross the path with precision. This interplay of awareness and calculated decision-making ensures a seamless navigation experience.



(a) The ego vehicle halts in front of a line of fast-moving vehicles.



(b) The ego vehicle successfully identifies a gap within the vehicle queue and proceeds.

Figure 11: Vehicle crossing scenario: Adaptive decision-making enables the ego vehicle to pause before a line of fast-moving vehicles and seamlessly identify a suitable gap within the queue, ensuring smooth and safe crossing.

1.6 INRIA's Zoé Autonomous driving platform

The tests have been conducted with the Inria's Renault Zoé autonomous vehicle, shown in Fig. 12, which is equipped with a Velodyne HDL-64 on the top, 3 Ibeo Lux LiDARs on the front and 1 on the back, Spectra SP90 RTK Dual antenna GNSS, Xsens IMU providing vehicle velocity and orientation, a stereo camera and 2 IDS cameras. Data from LiDARS are fused and synchronized using the IBEO fusion box. The perception system described earlier has been implemented on a PC in the trunk of the car, equipped with a Nvidia Titan X GPU, while the previously described automation process has been integrated in the vehicle.

Figure 12: INRIA's Renault Zoé autonomous experimental platform.

1.7 Transpolis testing platform

The tests took place at the Transpolis testing facility at Les Fromentaux (Figs. 13 and 14). In particular, all the scenarios defined for this POC were reproduced on a long boulevard with an intersection in the City area (Fig. 14).

- **City Area**: The City Area is a meticulously designed urban testing ground spanning 30 hectares. It features an intricate network of streets covering 12 kilometers, including two prominent boulevards with six lanes each. The area is divided into four sections, each presenting a unique layout with intersections, crossroads, and parking slots.

  To cater to diverse transportation needs, dedicated bus and cycle lanes have been incorporated. A ring road provides seamless access with three traffic lanes and four access lanes. The City Area boasts 40 real buildings, enabling connectivity testing in both line-of-sight and non-line-of-sight conditions.

  The infrastructure is equipped with adjustable facilities like fiber optic cabinets, roving sidewalk configurations, EV charging stations, and a dynamic changing-message sign. Movable signs, traffic lights with GLOSA services, and roundabouts cater to multifaceted testing requirements. Road markings, including luminescent lanes, provide precise guidance.

  The driving environments include varied surfaces, vegetation, and sloping terrain for comprehensive assessment. Spanning 7000 square meters, the City Area serves as a parking facility and event area for flexible usage. This technologically equipped urban proving ground showcases a commitment to advancing urban mobility.

The main boulevard intersection, where the experiment took place, is a 6 by 6 lanes intersection. The Zoé crosses the intersection on the lanes oriented West to East while the virtual obstacles cross the intersection on the lanes oriented North to South. The static obstacle most visible by the LiDARs are four buildings (one at each corner of the intersection), traffic lights and signs and 2 meters hight concrete panels installed on the north side of the road taken by the car.

The conducted experiments aimed to evaluate the behavior and performance of an ego-vehicle under various scenarios, focusing on autonomous control, collision avoidance, and response to different obstacles. The experiments were conducted on three different dates: 2023-02-21, 2023-02-22, and 2023-02-23.

Figure 13: Transpolis, headquartered at Les Fromentaux, is a cutting-edge testing ground for future urban mobility, where vehicles and infrastructures undergo daily trials with advanced equipment and technologies the[1].

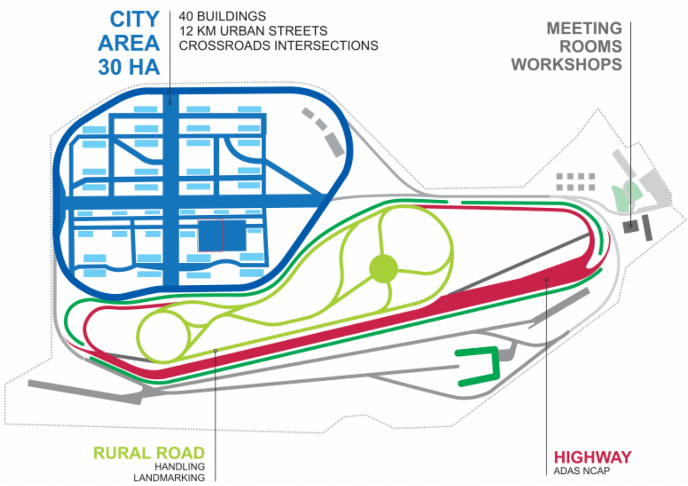


Figure 14: Les Fromentaux: Transpolis's areas [2]

1.8 Experiments at Transpolis and Results

During the experiments on 2023-02-21, the ego-vehicle demonstrated both stationary and autonomously controlled behaviors. While some scenarios involved non-operational situations with no movement, others showcased the ego-vehicle's ability to navigate autonomously with forward and backward motion. Collision avoidance maneuvers with buses, occluded cars, and fire trucks were observed, emphasizing the ego-vehicle's capability to detect and avoid potential collisions. The experiments also indicated that the ego-vehicle operated at a slower pace in certain scenarios, suggesting cautious navigation in complex environments.

The experiments on 2023-02-22 focused on the autonomous control of the ego-vehicle and its collision avoidance capabilities. Aggressive maneuvers were observed in some scenarios to avoid collisions with buses, resulting in deviations from the desired path. However, collision avoidance with trucks, occluded cars, and buses was successfully performed. The ego-vehicle demonstrated the ability to maintain a safe distance from obstacles while adjusting its trajectory to avoid collisions. Furthermore, the ego-vehicle exhibited responsiveness to the slow movement of virtual vehicles, enabling it to make informed decisions.

The experiments on 2023-02-23 continued to explore the ego-vehicle's autonomous control and collision avoidance abilities. The ego-vehicle exhibited forward and backward motion, employing emergency braking and maneuvering to avoid collisions with buses, fire trucks, and occluded cars. The effectiveness of collision avoidance strategies was evident, although challenges were observed in detecting and responding to hidden or obstructed objects. The experiments also highlighted the impact of speed on collision outcomes, emphasizing the importance of appropriate speed management in autonomous driving systems.

1.9 Data scenarios: Zoe autonomous vehicle + Dynamic Augmented obstacles

Several experiments were performed with the Zoe autonomous vehicle and the dynamic augmented obstacles. These augmented vehicles include a bus, a fire truck and a car. The bus and the fire truck can be recognized in all scenes, while the car is an occluded vehicle in many scenes. Throughout these experiments different setup was made to observe the interaction behavior between the ego-vehicle and augmented vehicles which include:

- Changing perception module parameters such as occupancy grid size generated by CMCDOT and threshold values for obstacle detection.
- Making changes to Model Predictive Controller (MPC) and prediction collision detector related to the navigation module.
- Modifying local planner target velocity of the ego-vehicle, target velocities of augmented vehicles, positions and timings of augmented vehicles.

By making these type of variations several interesting observations and interactions between the ego-vehicle and the augmented vehicles have been observed.

The data corresponding to several interaction scenarios has been collected in the form of rosbags. The experiments were performed for 3 consecutive days and the volume of total data collected during each day is displayed below:

- 2023_02_21 data: 351.5 GB
- 2023_02_22 data: 507.7 GB

Topic name	Topic type	Description
/zoe/velodyne_points	sensor_msgs/PointCloud2	Point clouds of the Velodyne HDL-64 LiDAR
/zoe/lux_right /zoe/lux_center /zoe/lux_left /zoe/lux_rear	sensor_msgs/PointCloud2 sensor_msgs/PointCloud2 sensor_msgs/PointCloud2 sensor_msgs/PointCloud2	Point clouds of the front right, front center, front left and rear Ibeo Lux LiDARs
/temp/zoe/velodyne_packets	velodyne_msgs/VelodyneScan	Raw data measurements from the Velodyne HDL-64
/zoe/classified_cloud	sensor_msgs/PointCloud2	Merged point cloud from the 5 LiDARs with classification of ground
/zoe/us_right /zoe/us_center /zoe/us_left	sensor_msgs/Range sensor_msgs/Range sensor_msgs/Range	Front ultrasonic range sensors
/zoe/sp90_fix /zoe/sp90_time_reference /zoe/fix /zoe/fix_common /zoe/raw_fix	sensor_msgs/NavSatFix sensor_msgs/TimeReference sensor_msgs/NavSatFix gps_common/GPSFix sensor_msgs/NavSatFix	Satellite localization of the Zoé
/zoe/camera_front/image_rect_color	sensor_msgs/Image	Images stream of the front camera
/zoe/camera_front/camera_info	sensor_msgs/CameraInfo	Information about the camera and its calibration
/zoe/imu/mag	sensor_msgs/MagneticField	Magnetic compass of the Zoé IMU
/zoe/imu/data	sensor_msgs/Imu	IMU data (orientation, angular velocity and linear acceleration)
/navigation/dwa_result	dwa_dynamic_planner/Trajectory	Current trajectory of the Zoé generated by the local planner
/navigation/planner_result /navigation/planner_status	dwa_dynamic_planner/PlannerResult dwa_dynamic_planner/PlannerStatus	Status information on the local planner
/zoe/velocity_grid	e_motion_perception_msgs/VelocityGrid	Grid of velocity vectors of the dynamic cells
/zoe/state_grid	e_motion_perception_msgs/FloatOccupancyGrid	Grid of filtered probability of occupied, dynamic, static and unknown
/zoe/occ_grid	e_motion_perception_msgs/FloatOccupancyGrid	Grid from one LiDAR point cloud of probabilities of occupied and unknown. Output of the LiDAR sensor model
/zoe/control/refs	ros_zoe_msgs/ControlRefs	Throttle, brake and steering commands sent to the hardware controller of the Zoé for automated driving
/tf /tf_static	tf2_msgs/TFMessage tf2_msgs/TFMessage	Dynamic and static transforms of the frames of the Zoé
/zoe/velocity /zoe/speed	geometry_msgs/TwistStamped geometry_msgs/TwistStamped	Velocity of the Zoé
/zoe/pose	geometry_msgs/PoseWithCovarianceStamped	Filtered Pose of the Zoé by a Kalman filter. Relative to a *world* fixed frame
/gazebo/set_model_state /gazebo/link_states /gazebo/model_states	gazebo_msgs/ModelState gazebo_msgs/LinkStates gazebo_msgs/ModelStates	States and status of the virtual Actors in Gazebo
/gazebo_scenario/rosparam	std_msgs/String	JSON serialization of all ROS parameters of the Zoé
/gazebo_scenario/scenario	std_msgs/String	JSON serialization of the scenario description and parameters

Table 1: Topics recorded during the experiment using the tool rosbag.

- 2023_02_23 data: 902.1 GB
- Total data: 1761.3 GB

1.9.1 ROS description

The software architecture of the Zoé has been designed using the robotic framework ROS (Robot Operating System), under version *melodic*. When using ROS, the software components of a robotic system are separated in nodes that communicate by sending messages on typed topic. For example, the LiDAR driver sends point cloud messages on the lidar topic, then the CMCDOT node listen to this topic to read the LiDAR measurements and it publishes its output grids on their respective topics.

To record and store the data from the experiment, we used the tool rosbag designed for ROS. Firstly, The rosbag recorder listen to a list of requested topics and read every messages sent (i.e. messages exchanged by the ROS nodes such as the LiDAR driver, CMCDOT node, GPS driver, local planner node). It stores a timestamped serialization of each message in a binary file called bag. In a second time, the rosbag player can read the messages stored in the rosbag and publish them on their respective topics to replay the recording. The player uses the timestamp of the messages to publish them in order and at the right simulated time. Table 1 shows a list of the topics recorded during the experiments.

The table below provides the details of each rosbag providing an overview of the scenario in general, the behavior of the ego-vehicle and its interaction with the augmented vehicles.

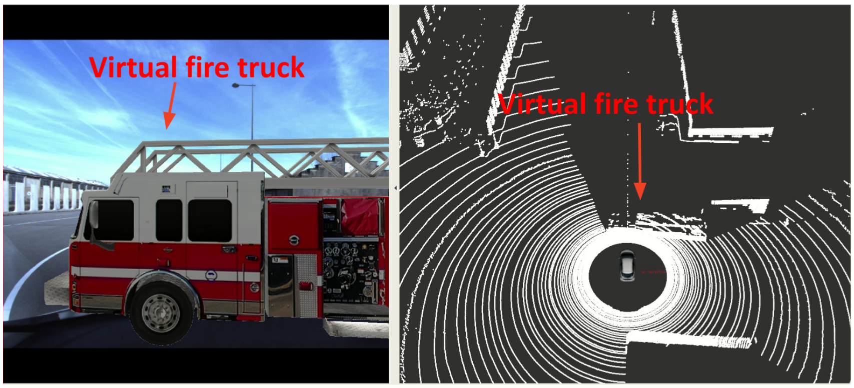
1.9.2 Experiments conducted on 2023_02_21

The first set of experiments showcased different aspects of the ego-vehicle's behavior. In some scenarios, the ego-vehicle remained stationary without any motion, indicating non-operational situations. However, in other cases, the ego-vehicle was autonomously controlled, demonstrating forward and backward motion. The descriptions of all the scenarios conducted on this day is provided in Table 2.

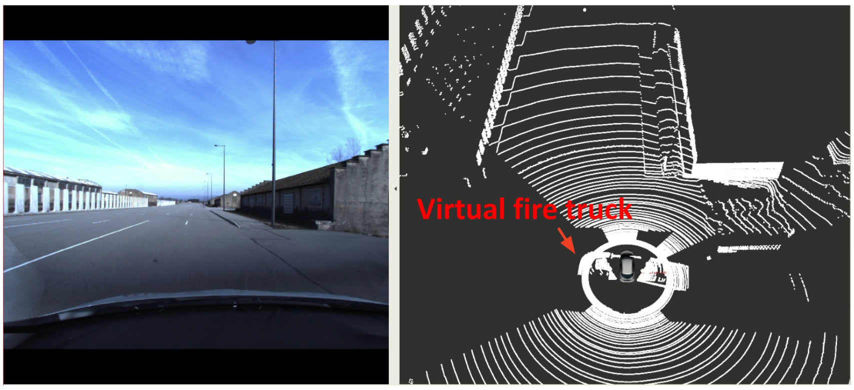
During autonomous control, the ego-vehicle exhibited collision avoidance maneuvers with various objects such as buses, occluded cars, and fire trucks. While some maneuvers led to collision with occluded car as shown in Fig. 16. showcased the vehicle's ability to detect potential collisions and take appropriate actions to avoid them. It is worth noting that the ego-vehicle moved at a slower pace in some scenarios, possibly indicating cautious navigation in complex environments.

The second set of experiments continued to explore the autonomous control of the ego-vehicle, focusing on collision avoidance with buses, trucks, and occluded cars. The ego-vehicle's ability to detect and respond to potential collisions was demonstrated through emergency braking and maneuvering as shown in Fig. 17 and 15.

Overall, the experiments conducted on February 21, 2023, provided valuable insights into the performance of the ego-vehicle under different scenarios. The results highlighted its capabilities in collision avoidance, emergency braking, and autonomous navigation.



(a) Ego-vehicle halts before the fire truck waiting for the vehicle to pass.



(b) The speed of the fire truck is slow, nevertheless ego-vehicle moves forward in the presence of this vehicle leading to collision.

Figure 15: Scenario from inria_zoe_2023_02_21_2: Ego-vehicle initially stops before the truck, but since the speed of truck is slow ego-vehicle then collides with it.

rosbag name	Description
inria_zoe_2023_02_21_15_41_52	The ego-vehicle remains stationary without any motion in this scenario.
inria_zoe_2023_02_21_15_42_24	The ego-vehicle is autonomously controlled and exhibits forward and backward motion. It also showcases collision avoidance maneuvers with a bus, an occluded car, and a fire truck. The ego-vehicle moves at a slow pace during this scenario.
inria_zoe_2023_02_21_15_52_42	The autonomous control of the ego-vehicle with forward and backward motion. However, in this case, the ego-vehicle moves at a moderate target velocity. The scenario includes emergency braking and collision avoidance with a bus. Notably, the fire truck moves at high speed and effortlessly avoids a collision with the ego-vehicle.
inria_zoe_2023_02_21_16_11_53	The ego-vehicle remains stationary without any motion in this scenario.
inria_zoe_2023_02_21_16_16_20	The ego-vehicle remains stationary without any motion in this scenario.
inria_zoe_2023_02_21_16_20_58	This scenario showcases autonomous control of the ego-vehicle with forward motion. It includes emergency braking and collision avoidance with a bus.
inria_zoe_2023_02_21_16_25_47	The ego-vehicle is autonomously controlled with forward motion in this scenario. Notably, there are no virtual vehicles present in the scene during this situation.
inria_zoe_2023_02_21_16_27_08	This scenario captures autonomous control of the ego-vehicle with both forward and backward motion. Similar to the previous scenario, there are no virtual vehicles present in the scene.
inria_zoe_2023_02_21_16_28_48	The ego-vehicle is autonomously controlled with only forward motion in this scenario. However, the virtual vehicles in the scene do not move.
inria_zoe_2023_02_21_16_31_52	This features autonomous control of the ego-vehicle with only forward motion. However, there are no virtual vehicles present in the scene.
inria_zoe_2023_02_21_1	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. The ego-vehicle performs collision avoidance maneuvers with a bus and a truck. However, a collision occurs with a car due to the car's low target velocity for this specific scene.
inria_zoe_2023_02_21_2	Ego-vehicle is autonomously controlled with forward and backward motion. It performs collision avoidance with a bus by emergency braking. Initially, it attempts to avoid a collision with a truck, but since the truck's speed is slow, the ego-vehicle moves forward and collides with the truck.
inria_zoe_2023_02_21_3	This scenario involves autonomous control of the ego-vehicle, with slight movement observed.
inria_zoe_2023_02_21_4	The ego-vehicle is autonomously controlled with forward motion in this scenario. It performs collision avoidance with a bus and an occluded car.
inria_zoe_2023_02_21_5	This features autonomous control of the ego-vehicle with forward motion. It successfully avoids a collision with a bus but collides with an occluded car.
inria_zoe_2023_02_21_6	Ego-vehicle is autonomously controlled with both forward and backward motion. It performs collision avoidance with a bus and an occluded car. Additionally, it successfully avoids a collision with a truck.
inria_zoe_2023_02_21_7	This scenario involves autonomous control of the ego-vehicle with forward motion. The ego-vehicle performs collision avoidance with a bus but collides with an occluded car.

Table 2: Description of rosbags for experiments conducted on 2023_02_21

1.9.3 Experiments conducted on 2023_02_22

The descriptions of the experiments corresponding to this day has been provided in Table 3. In the first set of experiments, the ego-vehicle was autonomously controlled with forward and backward motion. The model predictive controller exhibited aggressive actions as seen in Fig. 18, forcing the ego-vehicle away from a bus to avoid a collision. However, these aggressive maneuvers made it difficult for the ego-vehicle to follow the desired path. Additionally, due to the slow target velocity, there was no interaction between the ego-vehicle and a truck during backward motion.

In subsequent experiments, the ego-vehicle demonstrated collision avoidance with various obstacles such as a bus, a fire truck, and an occluded car. The ego-vehicle showcased the ability to halt and maintain a safe distance from the bus and the truck. It also adjusted its trajectory to avoid collisions while maintaining a safe distance from the fire truck. In some scenarios, the fire truck or virtual vehicles moved slowly, allowing the ego-vehicle to detect their presence and make appropriate decisions.

Overall, the experiments conducted on February 22, 2023, provided insights into the behavior and performance of the ego-vehicle under autonomous control. The results highlighted the vehicle's ability to perform collision avoidance maneuvers, adjust its trajectory, and maintain a safe distance from other objects on the road.

1.9.4 Experiments conducted on 2023_02_23

In the first set of experiments, where the ego-vehicle was autonomously controlled, it demonstrated both forward and backward motion. The model predictive controller employed aggressive maneuvers to keep the vehicle away from a bus and avoid collision as seen in (Fig. 22 and 23). However, these aggressive actions had a trade-off, as they made it challenging for the ego-vehicle to follow the desired path. Furthermore, due to the slow target velocity of the ego-vehicle (Fig. 25), there was no interaction observed between the vehicle and a truck during backward motion. Whereas in Fig. 19 the ego-vehicle avoids collision by applying emergency brakes. Sometimes ego-vehicle had just appropriate velocity to pass through the obstacles without any risk as seen in Fig. 20 and 26. All experiments are mentioned in Table 4 and 5.

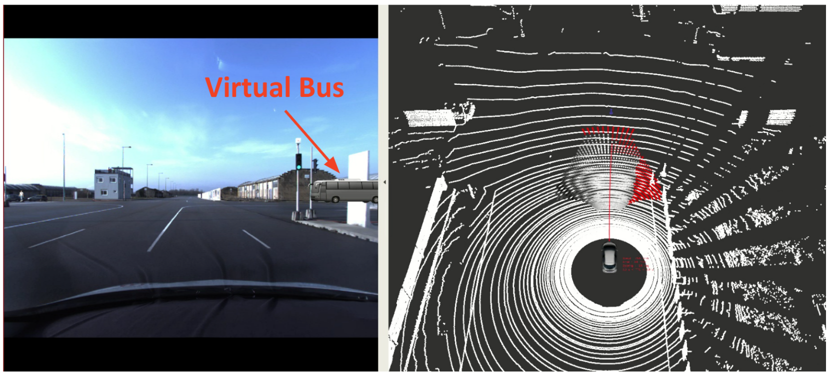
Moving on to the second set of experiments, the ego-vehicle exhibited autonomous control with forward and backward motion. Collision avoidance was a key aspect in these scenarios, and the ego-vehicle successfully avoided collisions with buses and fire trucks Fig. 24. However, it experienced collisions with occluded cars, indicating the complexity of detecting and responding to hidden or obstructed objects. The interaction between the ego-vehicle and the virtual vehicles varied depending on their speeds, leading to different collision outcomes as seen in Fig. 21.

In the final set of experiments, the ego-vehicle was subjected to manual control for its forward and backward motion. During these scenarios, the vehicle managed to avoid a collision with a bus. However, an unexpected increase in speed resulted in a collision with a barrier bus as seen in Fig. 27, highlighting the importance of maintaining appropriate speeds for safe driving. These experiments also included instances where the ego-vehicle was autonomously controlled with forward motion only, without any reported collisions or avoidance maneuvers.

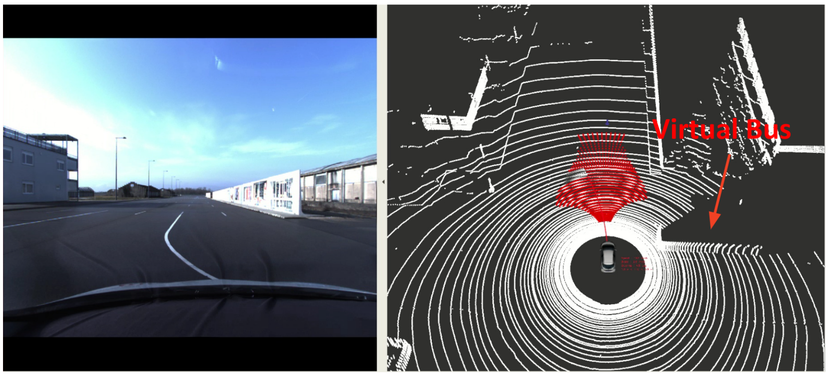
Overall, the experiments showcased the strengths and limitations of the ego-vehicle's autonomous and manual control systems. They highlighted the effectiveness of collision avoidance strategies in certain scenarios, as well as the challenges posed by occluded or hidden objects. Additionally, the influence of speed on collision outcomes was evident, emphasizing the need for appropriate speed management in autonomous driving systems. These findings contribute to the ongoing efforts in improving the safety and reliability of autonomous vehicles, further refining collision detection and avoidance algorithms, and enhancing overall driving performance.

rosbag name	Description
inria_zoe_2023_02_22_1	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. The actions from the model predictive controller are quite aggressive, forcing the ego-vehicle away from a bus to avoid collision. However, the aggressive maneuvers make it difficult for the ego-vehicle to follow the desired path. The target velocity of the ego-vehicle is slow, resulting in no interaction between the truck and the ego-vehicle during backward motion.
inria_zoe_2023_02_22_2	This scenario features autonomous control of the ego-vehicle with forward and backward motion. Further details about this scenario are not provided in the table.
inria_zoe_2023_02_22_3	In this scenario, the ego-vehicle is autonomously controlled with forward and backward motion. It performs collision avoidance with a bus and a fire truck, given the low target velocity of the ego-vehicle. Due to the low speed, there is no interaction between the ego-vehicle and the truck.
inria_zoe_2023_02_22_4	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. The ego-vehicle halts to avoid a collision with a bus. Due to the low speed, the ego-vehicle does not have any interaction with the fire truck, eliminating the possibility of a collision.
inria_zoe_2023_02_22_5	Ego-vehicle is autonomously controlled with forward and backward motion. The ego-vehicle moves slowly and stops at a safe distance from the bus and truck.
inria_zoe_2023_02_22_6	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. The ego-vehicle stops at a safe distance from the bus while having limited interaction with the truck due to its low speed.
inria_zoe_2023_02_22_7	Ego-vehicle is autonomously controlled with forward and backward motion. The speed of the bus is slow, resulting in no interaction with the ego-vehicle. The ego-vehicle successfully avoids collision with the fire truck.
inria_zoe_2023_02_22_8	This scenario features autonomous control of the ego-vehicle with forward and backward motion. The ego-vehicle avoids collision with a bus by slightly deviating from its normal path. However, it collides with an occluded car during this process. The fire truck does not move in this scenario.
inria_zoe_2023_02_22_9	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. The ego-vehicle performs collision avoidance with a bus but collides with an occluded car. The fire truck moves very slowly, and the ego-vehicle detects this, waits until the truck has passed, and then proceeds forward while maintaining a safe distance.
inria_zoe_2023_02_22_10	The ego-vehicle remains stationary without any movement in this scenario.
inria_zoe_2023_02_22_11	Ego-vehicle is autonomously controlled with forward motion at a moderate speed. The ego-vehicle avoids collision with a bus by increasing its speed but later collides with an occluded car.
inria_zoe_2023_02_22_12	This scenario involves autonomous control of the ego-vehicle. For forward motion, the virtual bus moves very slowly, so the ego-vehicle proceeds without paying attention to the bus. During backward motion, the virtual truck moves at a very low speed, allowing the ego-vehicle to continue its motion as usual.
inria_zoe_2023_02_22_13	Ego-vehicle is autonomously controlled. During forward motion, the bus moves very slowly, resulting in no interaction with the ego-vehicle. A similar situation occurs with the fire truck during the ego-vehicle's backward motion.

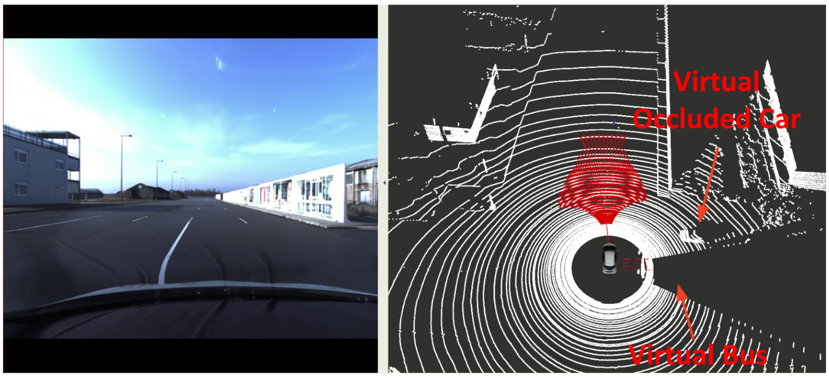
Table 3: Description of rosbags for experiments conducted on 2023_02_22



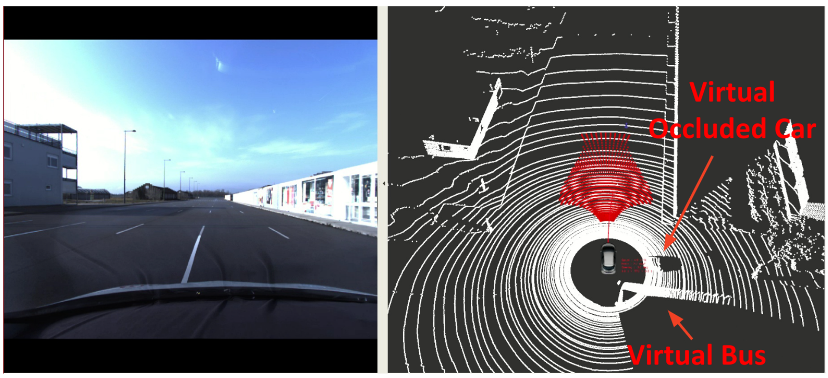
(a) MPC has predicted a collision with an augmented bus in the future



(b) Ego-vehicle slightly deviates from its path in order to avoid collision with bus



(c) An occluded car appears in the scene and approaches the ego-vehicle

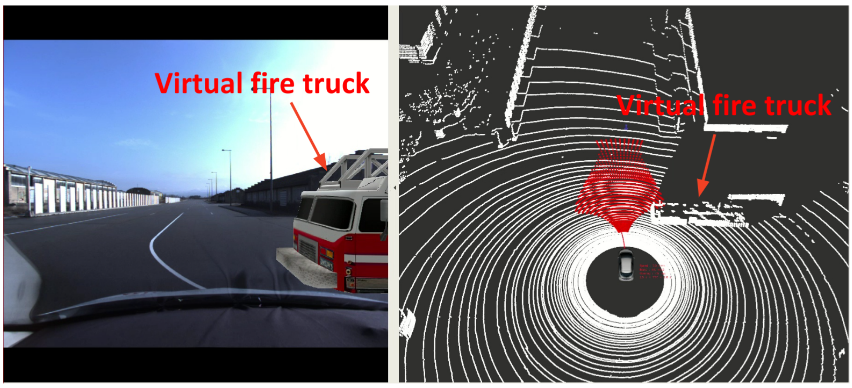


(d) Ego-vehicle cannot detect the occluded car and there is a collision

Figure 16: Scenario from inria_zoe_2023_02_23_10_21_41: Collision avoidance with bus leading to collision with an occluded car.

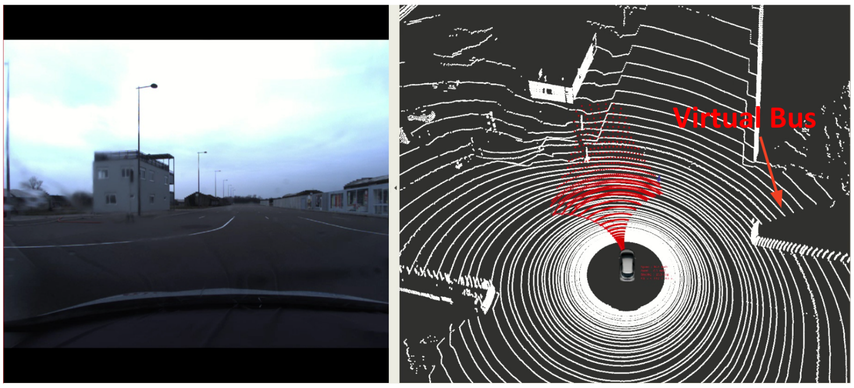
(a) MPC predicts a future collision

(b) Ego-vehicle approaches a fire truck

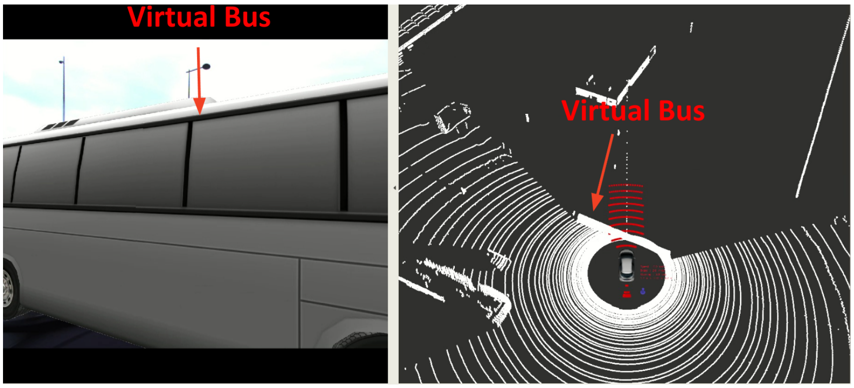


(c) Ego-vehicle applies emergency brakes but due to reasonable speed of fire truck, there is a collision

Figure 17: Scenario from inria_zoe_2023_02_23_10_21_41: Emergency braking in front of a fire truck and collision with it.



(a) MPC detects a virtual bus and forces quite aggressive actions for ego-vehicle



(b) MPC leads ego-vehicle to deviate alot from its default path, ego-vehicle stops for the bus to pass by

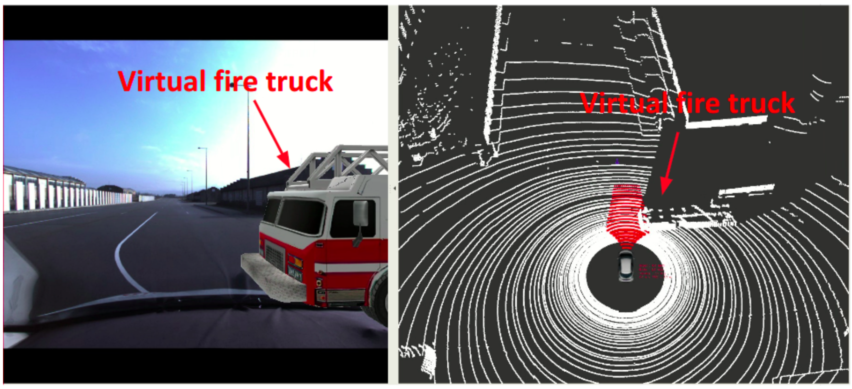
Figure 18: Scenario from inria_zoe_2023_02_22_1: Model predictive controller is quite aggressive forcing the ego-vehicle away from the bus and avoiding collision. Although the collision has been avoided, now it becomes difficult for the ego-vehicle to follow desired path.

rosbag name	Description
inria_zoe_2023_02_23_14_38_31	contrôle manuel
inria_zoe_2023_02_23_14_41_09	contrôle manuel
inria_zoe_2023_02_23_14_44_00	contrôle manuel
inria_zoe_2023_02_23_14_46_01	contrôle manuel
inria_zoe_2023_02_23_14_47_53	contrôle manuel
inria_zoe_2023_02_23_14_49_53	contrôle manuel
inria_zoe_2023_02_23_14_51_45	contrôle manuel
inria_zoe_2023_02_23_14_53_44	contrôle manuel
inria_zoe_2023_02_23_14_56_19	contrôle manuel
inria_zoe_2023_02_23_09_32_51	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. There are no collisions or collision avoidance maneuvers.
inria_zoe_2023_02_23_09_54_31	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. It performs collision avoidance with a bus and an occluded car. However, emergency braking results in a collision with a fire truck.
inria_zoe_2023_02_23_09_59_20	This scenario features autonomous control of the ego-vehicle with forward and backward motion. Emergency braking leads to collision avoidance with a fire truck, but a collision occurs with an incoming/occluded car.
inria_zoe_2023_02_23_10_03_00	This involves autonomous control of the ego-vehicle with forward and backward motion. Emergency braking leads to collision avoidance with a fire truck, but a collision occurs with an incoming/occluded car.
inria_zoe_2023_02_23_10_10_08	In this scenario, the ego-vehicle is autonomously controlled with only forward motion. Emergency braking and collision occur with a bus while an occluded car passes by.
inria_zoe_2023_02_23_10_12_21	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. Emergency braking and avoidance maneuvers lead to a collision with a fire truck.
inria_zoe_2023_02_23_10_15_33	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. Collision avoidance is performed to deal with the high speed of the ego-vehicle.
inria_zoe_2023_02_23_10_17_49	This scenario features autonomous control of the ego-vehicle with forward and backward motion. Collision avoidance with a bus leads to a collision with a car, followed by collision avoidance with a fire truck, resulting in a collision with the fire truck.
inria_zoe_2023_02_23_10_21_41	Ego-vehicle is autonomously controlled with forward and backward motion. Collision avoidance with a bus leads to a collision with a car. Emergency braking in front of a fire truck results in collision avoidance with the fire truck.
inria_zoe_2023_02_23_10_26_20	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. Collision avoidance with a bus is followed by emergency braking in front of a fire truck. However, the model predictive controller suggests actions towards the fire truck.
inria_zoe_2023_02_23_10_30_30	Ego-vehicle is autonomously controlled with forward and backward motion in this scenario. Collision avoidance with a bus is followed by the slowing down of a car. The model predictive controller suggests actions towards the fire truck, but the ego-vehicle slows down and avoids a collision.
inria_zoe_2023_02_23_10_35_15	This scenario features autonomous control of the ego-vehicle with forward and backward motion. Collision avoidance with a bus is followed by the slowing down of a car before a fire truck, resulting in a collision with the car.
inria_zoe_2023_02_23_10_40_51	Ego-vehicle is autonomously controlled with forward and backward motion. Collision avoidance with a bus is followed by the slowing down of a car before a fire truck, resulting in a collision with the car.
inria_zoe_2023_02_23_10_49_09	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. Emergency braking is performed in front of a bus, followed by emergency braking in front of a fire truck.

Table 4: Part 1: Description of rosbags for experiments conducted on 2023_02_23

rosbag name	Description
inria_zoe_2023_02_23_10_54_07	Ego-vehicle is autonomously controlled with forward and backward motion in this scenario. Collision avoidance with a bus is followed by emergency braking in front of a fire truck.
inria_zoe_2023_02_23_10_59_30	This scenario features autonomous control of the ego-vehicle with forward and backward motion. Collision avoidance with a bus is followed by emergency braking in front of a fire truck.
inria_zoe_2023_02_23_11_04_33	Ego-vehicle is autonomously controlled with forward and backward motion. A collision occurs with a car hidden behind a bus, followed by braking in front of a fire truck.
inria_zoe_2023_02_23_15_01_07	This scenario involves autonomous control of the ego-vehicle with forward and backward motion. There are no collisions or collision avoidance maneuvers observed.
inria_zoe_2023_02_23_15_16_49	In this scenario, manual control is used for the ego-vehicle's forward and backward motion. The ego-vehicle successfully avoids a collision with a bus. However, it later picks up speed and collides with the barrier bus due to the high speed.
inria_zoe_2023_02_23_15_21_06	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. There are no collisions or collision avoidance maneuvers observed.
inria_zoe_2023_02_23_15_23_31	This scenario involves autonomous control of the ego-vehicle with only forward motion. There is no collision with the bus, but a collision occurs with an occluded car.
inria_zoe_2023_02_23_15_25_53	The ego-vehicle is autonomously controlled with forward and backward motion in this scenario. It performs collision avoidance with a bus, followed by collision avoidance with a truck.
inria_zoe_2023_02_23_15_28_34	In this scenario, the ego-vehicle is autonomously controlled with only forward motion. There are no collisions or collision avoidance maneuvers observed.
inria_zoe_2023_02_23_15_30_54	This scenario involves autonomous control of the ego-vehicle with only forward motion. There are no collisions or collision avoidance maneuvers observed.
inria_zoe_2023_02_23_15_32_43	In this scenario, manual control is used for the ego-vehicle's forward and backward motion. Emergency braking is performed after encountering a truck.

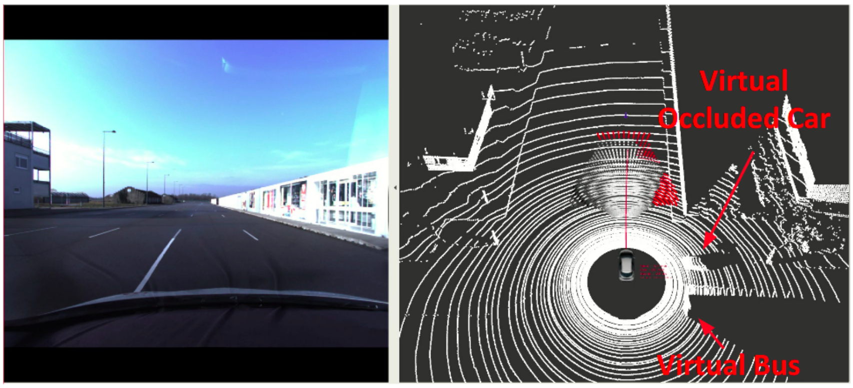
Table 5: Part 2: Description of rosbags for experiments conducted on 2023_02_23



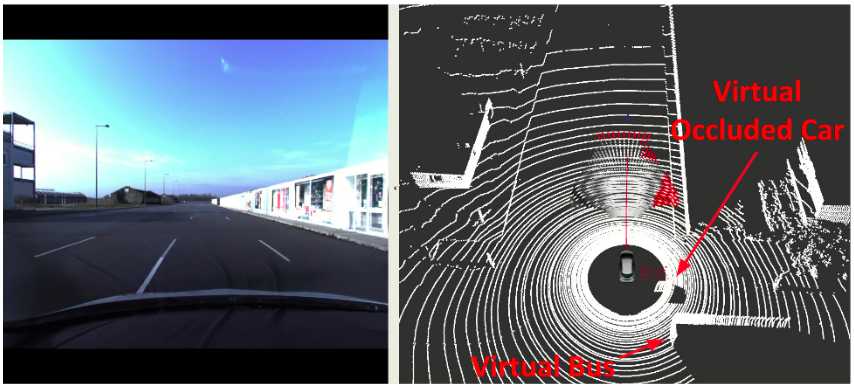
(a) Ego-vehicle applies emergency brakes after perceiving the fire truck

(b) Ego-vehicle stops, allowing the virtual truck to pass

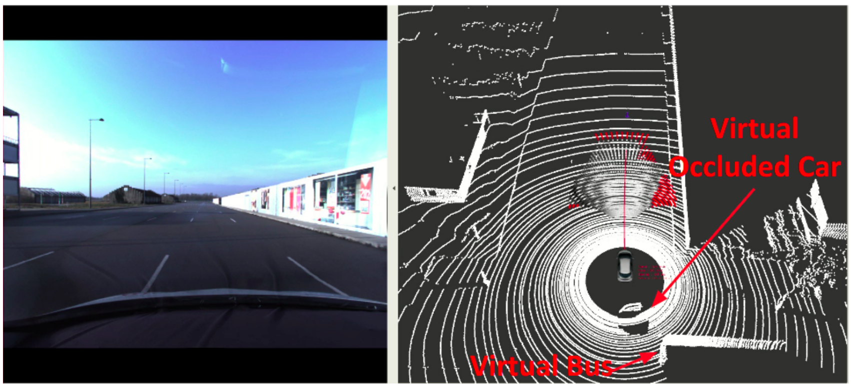
Figure 19: Scenario from inria_zoe_2023_02_23_09_59_20: Collision avoidance by applying emergency brakes in front of a fire truck.



(a) Potential risk of collision is detected by MPC, while the approaching occluded car was not detected by it

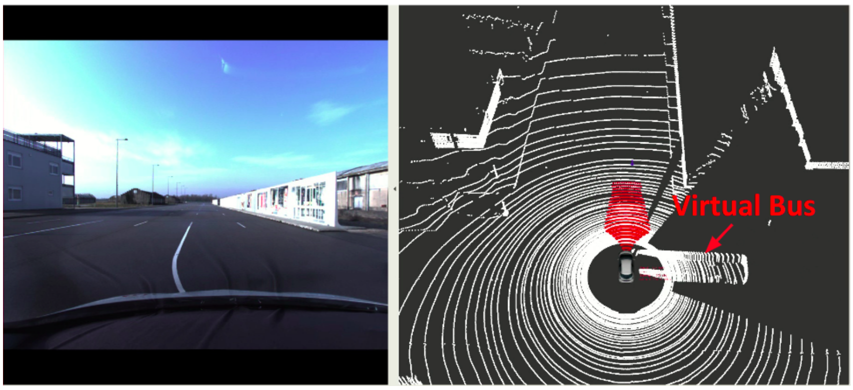


(b) The target velocity of ego-vehicle is just appropriate and it avoid a near-collision with the virtual car

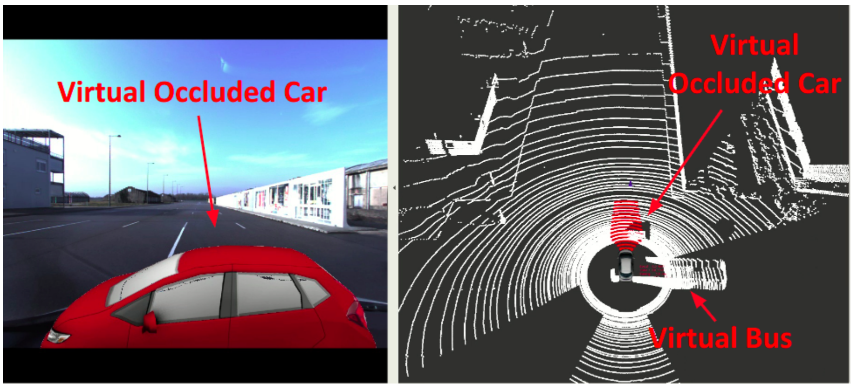


(c) Ego-vehicle overcome the risk of collision

Figure 20: Scenario from inria_zoe_2023_02_23_09_54_31: Collision avoidance with the bus and occluded car by having the appropriate target velocity for the ego-vehicle.

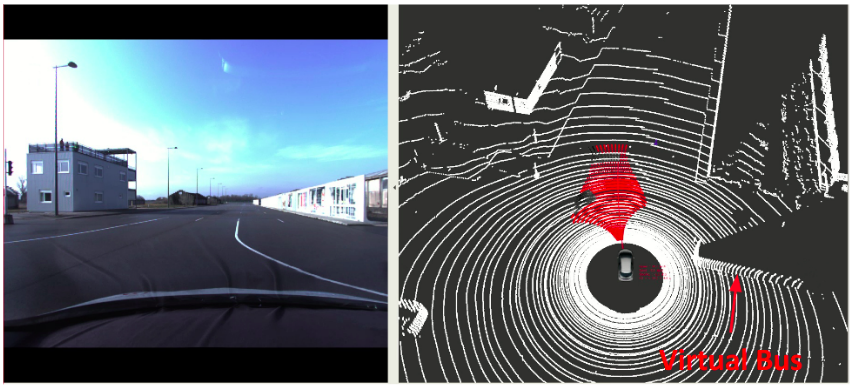


(a) MPC detected a virtual bus, but the ego-vehicle could not apply emergency brakes at the right position and so the virtual bus collides with ego-vehicle

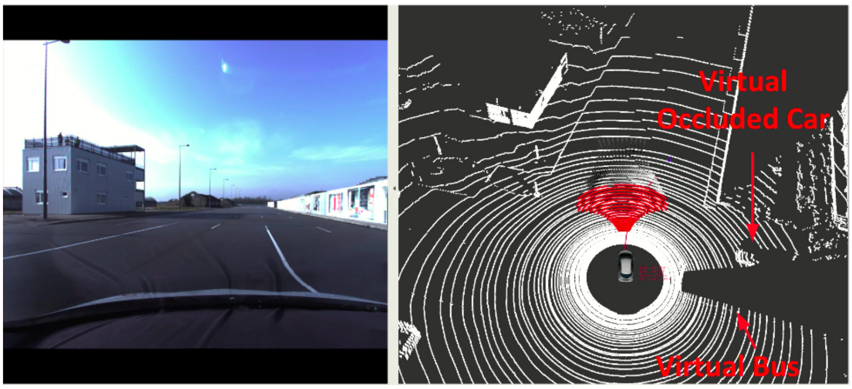


(b) A virtual occluded car is seen passing by illustrates that the MPC failed in stopping the ego-vehicle at a safety distance from the virtual bus

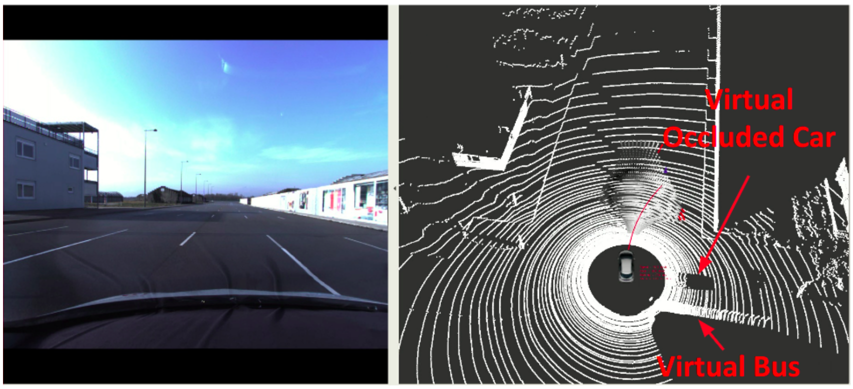
Figure 21: Scenario from inria_zoe_2023_02_23_10_10_08: Emergency braking and collision with bus while the occluded car passes by.



(a) MPC predicts potential risk with a virtual bus and drastically deviates the ego-vehicle form its path

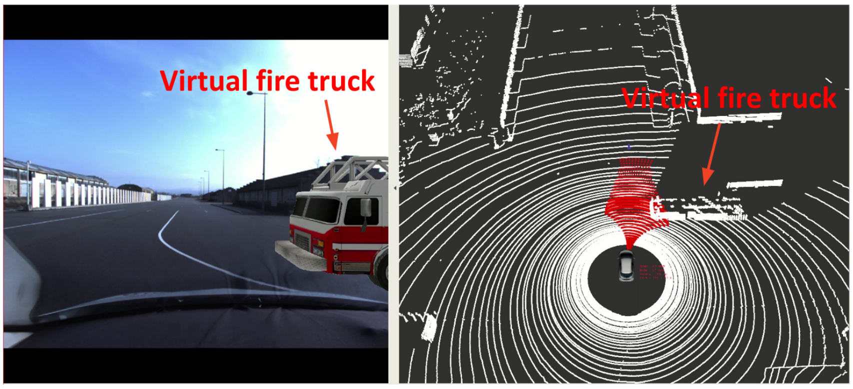


(b) Considering the velocities of augmented vehicle ego-vehicle tries to return to default path



(c) Ego-vehicle avoids collision and returns to assigned path

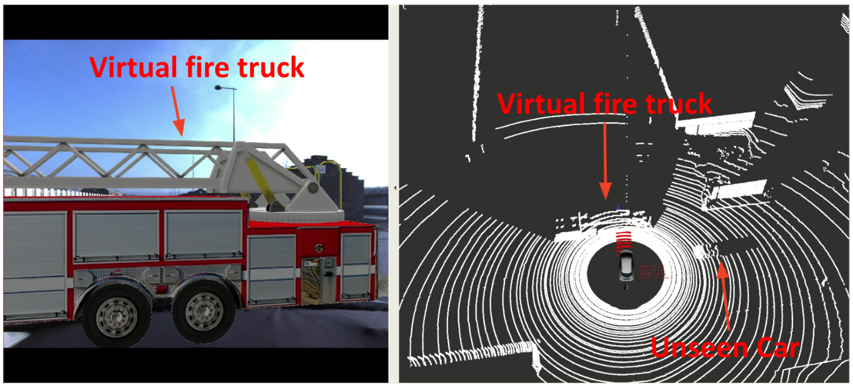
Figure 22: Scenario from inria_zoe_2023_02_23_10_17_49: Collision avoidance with the bus and occluded car by slight aggressive behavior by the model predictive controller for the ego-vehicle.



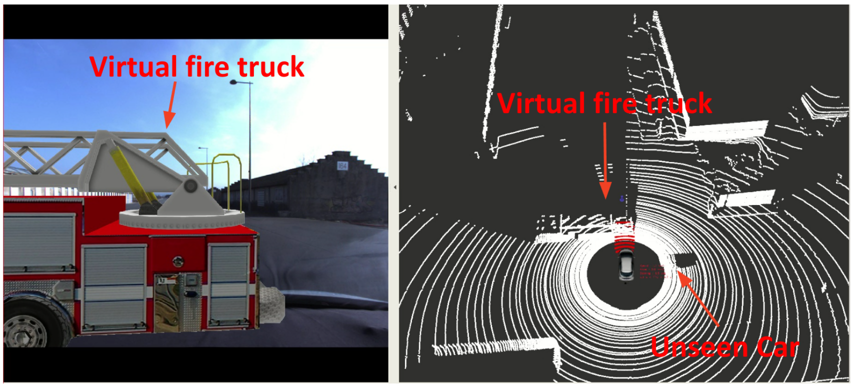
(a) MPC detects a risk of collision, but it takes motion actions towards the virtual fire truck and then applies emergency brakes

(b) Ego-vehicle stops after applying emergency brakes

Figure 23: Scenario from inria_zoe_2023_02_23_10_30_30: Ego-vehicle slows down to avoid collision with the fire truck, but in this case model predictive controller suggests actions towards the fire truck before the ego-vehicle fully stops. Similar behavior is also observed in Scenario inria_zoe_2023_02_23_10_35_15.

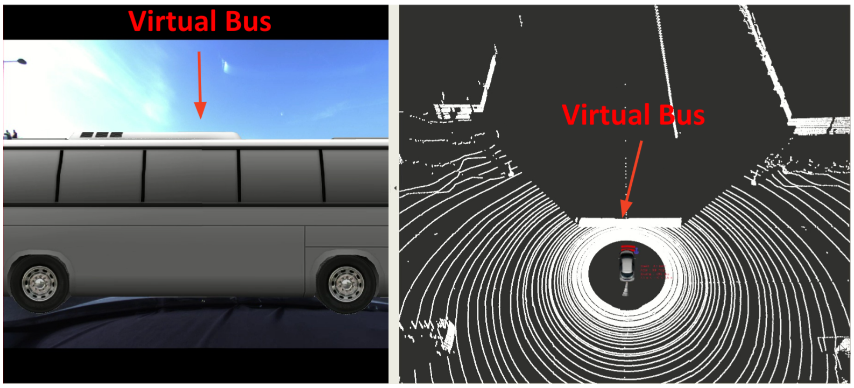


(a) Ego-vehicle halts allowing the fire truck to pass and suddenly an unseen virtual car approaching it

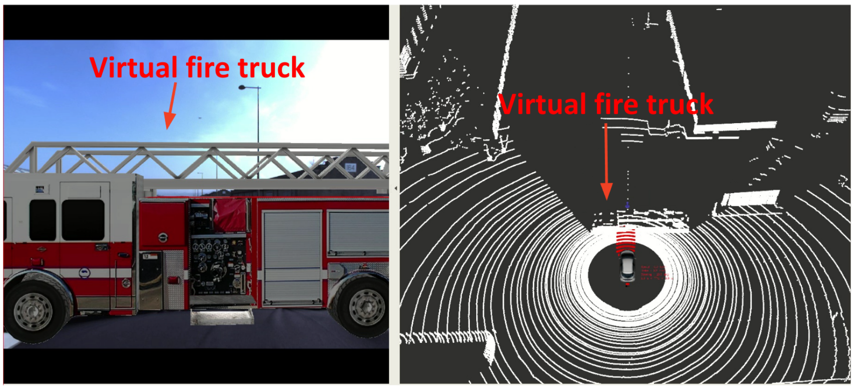


(b) The unseen car collides with ego-vehicle

Figure 24: Scenario from inria_zoe_2023_02_23_10_40_51: Ego-vehicle has stopped just before the fire truck to avoid collision. However, an unseen virtual vehicle collide with ego-vehicle during this interval.

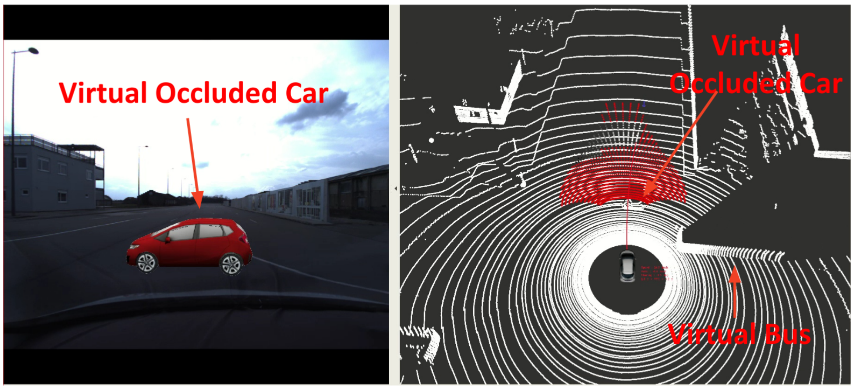


(a) Ego-vehicle stops at a safe distance from the virtual bus

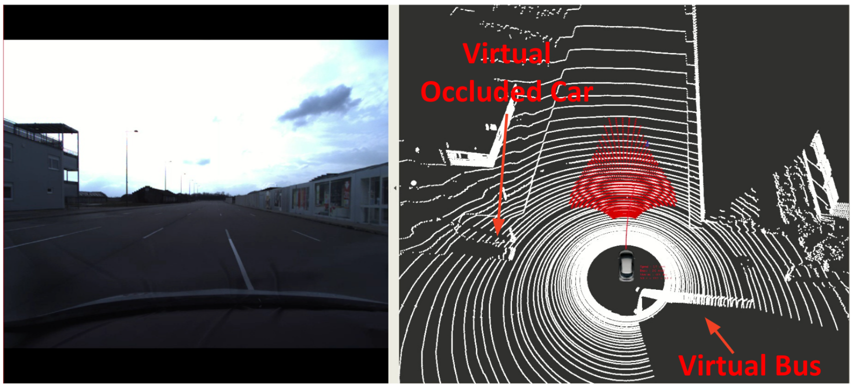


(b) Ego-vehicle stops at a safe distance from the virtual truck

Figure 25: Scenario from inria_zoe_2023_02_23_10_49_09: Ego-vehicle moves with low target velocity in this scenario. It is able to stop at a safe distance from the bus and the fire truck.

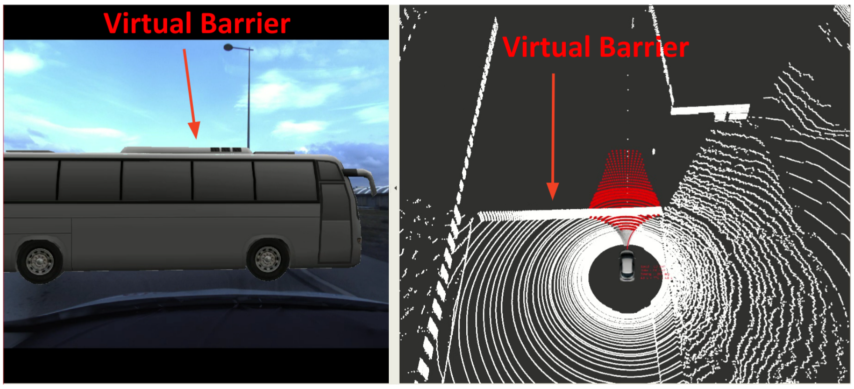


(a) Occluded car moves at higher velocity in this scene, safely passing by the ego-vehicle

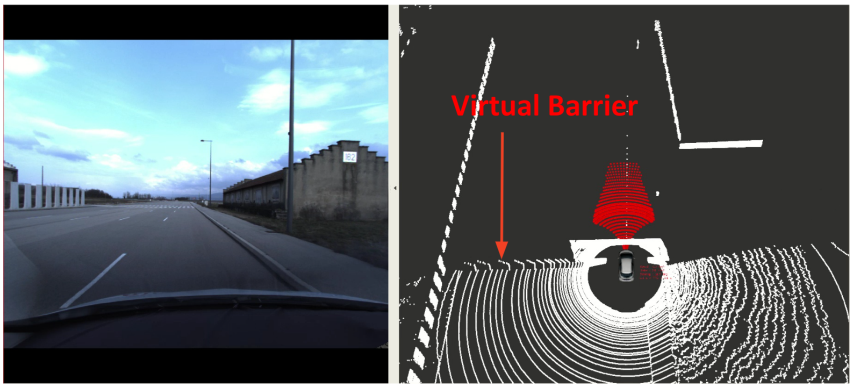


(b) Ego-vehicle's velocity is appropriate to avoid collision with virtual bus and car

Figure 26: Scenario from inria_zoe_2023_02_23_15_25_53: Ego-vehicle manages to avoid collision with bus and occluded car by having the accurate target velocity.



(a) Ego-vehicle approaches an virtual barrier in the form of a bus and MPC predicts a collision risk



(b) MPC fails to stop the ego-vehicle at accurate time and it breaks into the virtual barrier

Figure 27: Scenario from inria_zoe_2023_02_23_15_16_49: Ego-vehicle is unable to slow down and collides with the virtual barrier towards the end of the assigned path.

1.10 Future steps

Fig. 28 shows a 2D satellite view of the Transpolis testing facility. In the next steps of this POC, we plan to create a simulation setup by integrating this satellite view and localizing the dynamic augmented agents generated in Gazebo. The idea is to find correlation in the behavior of the ego-vehicle in simulation against the real-world data collected at Transpolis. In this way, it is possible to validate the repeatability of simulation, i.e, perception and navigation modules, to determine if these are deterministic by comparing the simulated data and real-world Transpolis data.

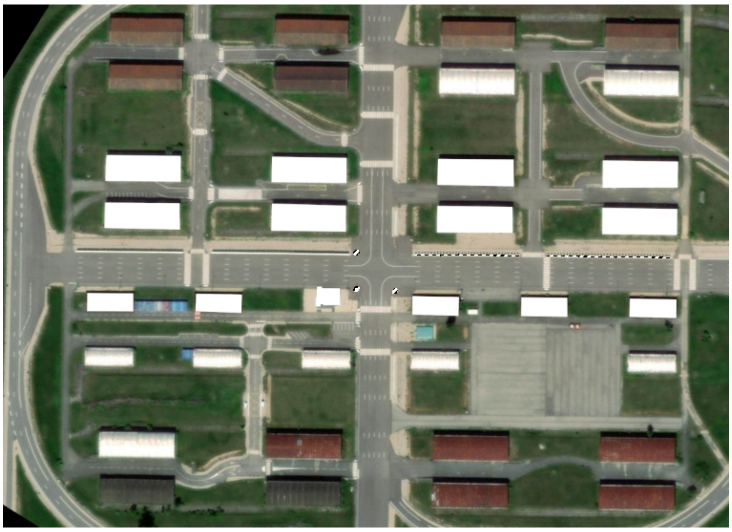


Figure 28: Generated ground truth using google map image to be integrated in simulation along with augmented vehicles.

For performing comparison between simulation and real-world data a new type of metric called PFC-MSE has been introduced in [8]. This metric evaluates the similarity between two occupancy grids by comparing the behavior of a navigation algorithm on the grids. Using the data from Transpolis experiments and a ground truth of the environment in the format of an occupancy grid we could evaluate any perception module, e.g. CMCDOT, by applying PFC-MSE on the CMCDOT output and the ground truth. Figure 29 shows how PFC-MSE evaluate the similarity of two occupancy grids. Given the large amount of data recorded in the experiment we aim to propose a comprehensive evaluation of occupancy grid based perception modules using PFC-MSE.

In future visit to Transpolis, we plan to perform more experiments with augmented agents considering different type of scenarios. So far the augmented agents where just vehicles, but we can include other road actors such as pedestrians. This will enhance the dataset to obtain more information and understand the behavior of the ego-vehicle subjected to these virtual agents. These more diverse scenarios will improve the relevance of the evaluation that can be performed using only the data from our first experiment.

(a) Ground Truth (GT) (b) GT cost grid (c) Inference (d) Inference cost grid (e) Distortion grid

Figure 29: Illustration of the evaluation of PFC-MSE an inference of an environment against its corresponding ground truth. 29a and 29b are the ground truth and its cost grid, 29c and 29d are the inference and its cost grid. Cost grids cell values are the navigation costs to the cells. Paths drawn in red the cost grids shows the differences of behavior of the navigation on the grids. The resulting distortion grid 29e is the pixel-wise absolute error between both cost grids, it is also weighed by the disjunctive probability of free occupancy on the GT or the inference. In this example the PFC-MSE value is $1.634e2$

2 CEREMA / LNE POC

2.1 Introduction

More and more intelligent systems on vehicles use AI (e.g. visual or mixed navigation, sign recognition, road tracking, obstacle detection). The qualification of these systems requires verification in all kinds of scenarios, including, for example, taking into account degraded weather conditions. For cost and safety reasons, these qualification tests cannot be carried out in real conditions, as some tests may present risks or have frequencies of occurrence too low to allow the collection of large series of data. For this reason, sensor simulation tools and degraded weather conditions (physical, numerical or hybrid) must be implemented. These simulation tools need to be validated and qualified. In particular, it is necessary to verify on them:

- The repeatability of a test on the same tool.
- The reproducibility of a test from one tool to another.

LNE and Cerema have different tools for AI systems evaluation at their disposal as shown on Figure 30, that need to be qualified:

- PAVIN fog and rain platform for producing artificial fog and rain [9].
- Cerema noise models for numerical simulation of fog (partial digital simulation) by using PAVIN Platform data as input initially without fog.
- LEIA platform for artificial intelligence evaluation:
  - Full digital simulation (sensor + weather) in LEIA 1.
  - Replay of videos recorded in LEIA 1 (pure simulation) in LEIA 2 (hybrid test using physical sensor).

In this study, we focus on the example of pedestrian detection, taking fog conditions into account. Different test methods are compared (see Figure 31), using the score obtained by the state-of-the-art YOLO algorithm as a metric.

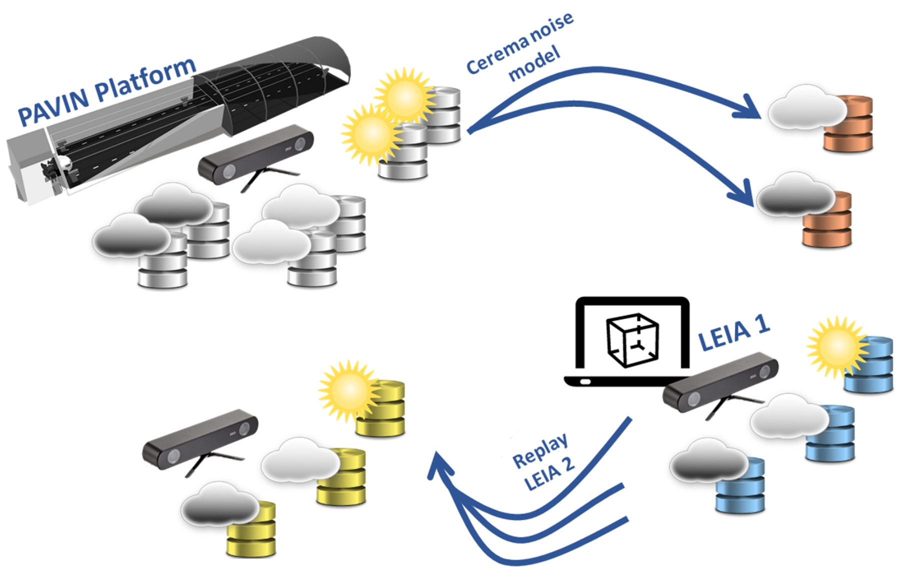


Figure 30: Principle of the method used to compare physical, numerical and hybrid tests.

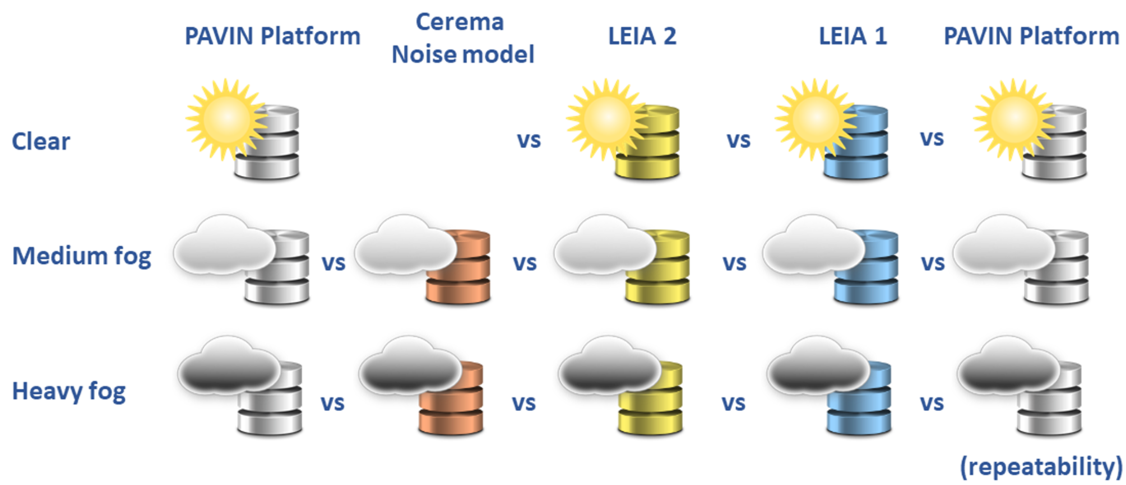


Figure 31: Comparison of physical, numerical and hybrid test data.

2.2 Tests realization

2.2.1 Physical tests

The objective of the tests scenarios defined by Cerema is to collect videos containing 100 individual pedestrians moving in a scene subjected to various weather conditions (clear weather and two types of fog), lighting conditions (day or night) and seasons using clothing representative of summer or winter. To ensure the repeatability of the measurements, each pedestrian's journey is made twice for each configuration of the scene, weather conditions and pedestrian clothing. The dataset of tests therefore includes a total of **2 runs x 100 pedestrians x 3 weather x 2 lighting = 1200 videos**.

The three types of weather conditions chosen are :

- *Clear weather* (CW): it allows to have a reference scene without disturbances due to the presence of fog.

- *Medium fog* (MF) : the visibility is of 23 m allowing to modify the general aspect of the objects of the scene by leaving detectable all the elements of the visible scene.

- *Dense fog* (DF): the visibility is of 10 m allowing elements of the background to disappear for stereo camera but not for thermal camera.

Different objects are placed in the scene to reproduce an urban scene. They remained in the same position for the duration of the tests to ensure reproducibility and to allow comparison of the datasets under different lighting and weather conditions. Here is a list of the objects used:

- Shrubs: A ficus in the background and a large planter with two shrubs in the left foreground.

- Wooden picnic table in the foreground right.

- Orange traffic cones (x3 positioned in line and at equal distance).

- Vehicle (Renault Megane).

- Some traffic signs (Speed limit 60, speed limit 50 and a wildlife crossing sign).

- Ground marking strips: crosswalk and dashed marking.

- Four calibrated targets (a large black and a large grey (50 x 50 cm), a small white and a small black (30 x 30 cm)).

Figure 32: Daytime scene of the PAVIN platform for the PRISSMA tests.

To add a seasonality in the scene (summer/winter), the pedestrians have been dressed with clothes characteristic of high or low temperatures such as: hats, caps, shorts, pants, coats, ... and as much as possible, a variability of the color of the clothes has been respected (bright colors, dark or light colors). Wigs have also been used to increase the number of female pedestrians. To break the pedestrian silhouette, accessories have been used to constrain the pedestrian detection algorithms : Balloon (soccer and rugby), backpack, computer shoulder bag, tote bag, hiking bag, walking sticks, open or closed umbrella, wooden board, cardboard box, snowboard, green plant, survival blanket, headlamp.

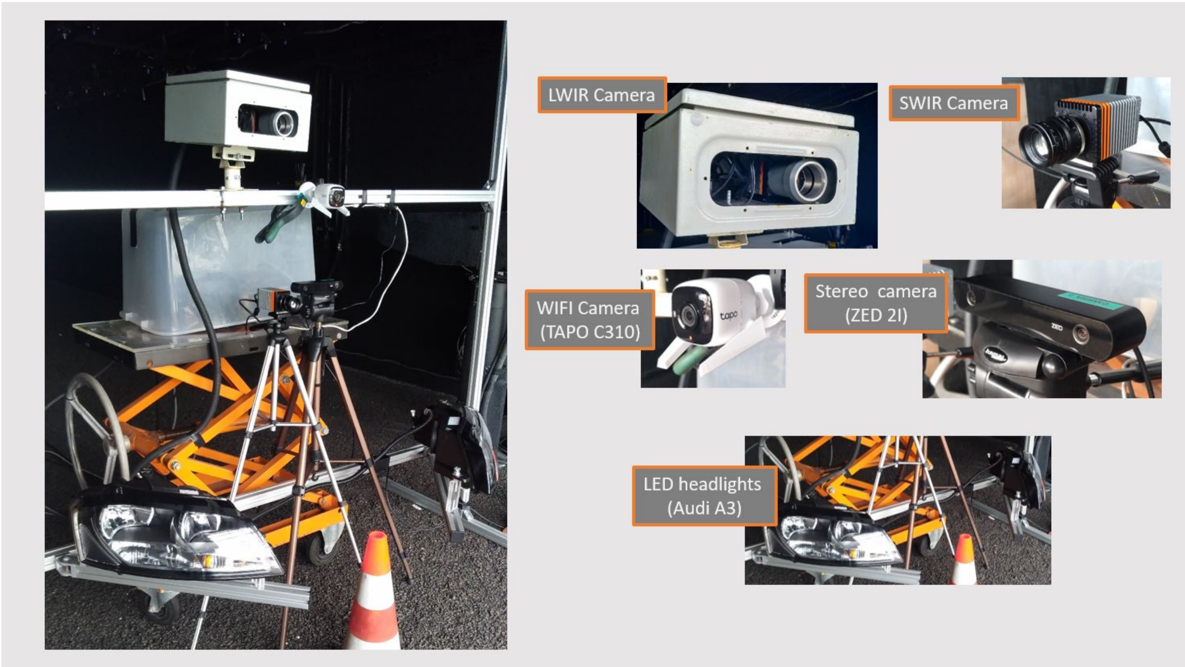


Figure 33: Instrument layout for PRISSMA tests

To obtain a well-characterized dataset, measurements are performed with the following sensors:

- Stereo camera (ZED 2i model) (depth and RGB channels of the image).
- WIFI camera (TAPO C310) (visible image of the scene).
- SWIR camera (Xenics).
- LWIR camera (Xenics).
- Weather sensors of the PAVIN platform (Temperature, humidity, visibility, ...).

The different instruments were positioned at the beginning of the greenhouse (See Figure 33). The bottom of the greenhouse has a transparent rectangular opening covered with a black cover during the night tests. This avoids glare caused by car headlights placed at instrument level.

The database includes 100 different pedestrians (clothes and accessories) as shown in Figure 34, moving along an identical route (see Figure 35) of a duration of approximately 1 minute depending on each pedestrian's walking pace. Each route is repeated twice to test reproducibility. Each pedestrian evolves in the two lighting conditions (day and night) and for the three weather conditions (clear weather, fog visibility of 10 m, fog visibility of 23 m), which corresponds to 2 x 2 x 3 x 100 = 1200 one-minute sequences, i.e. nearly 20 hours of testing. This is not including additional time, such as change of weather conditions (Clear / Fog).

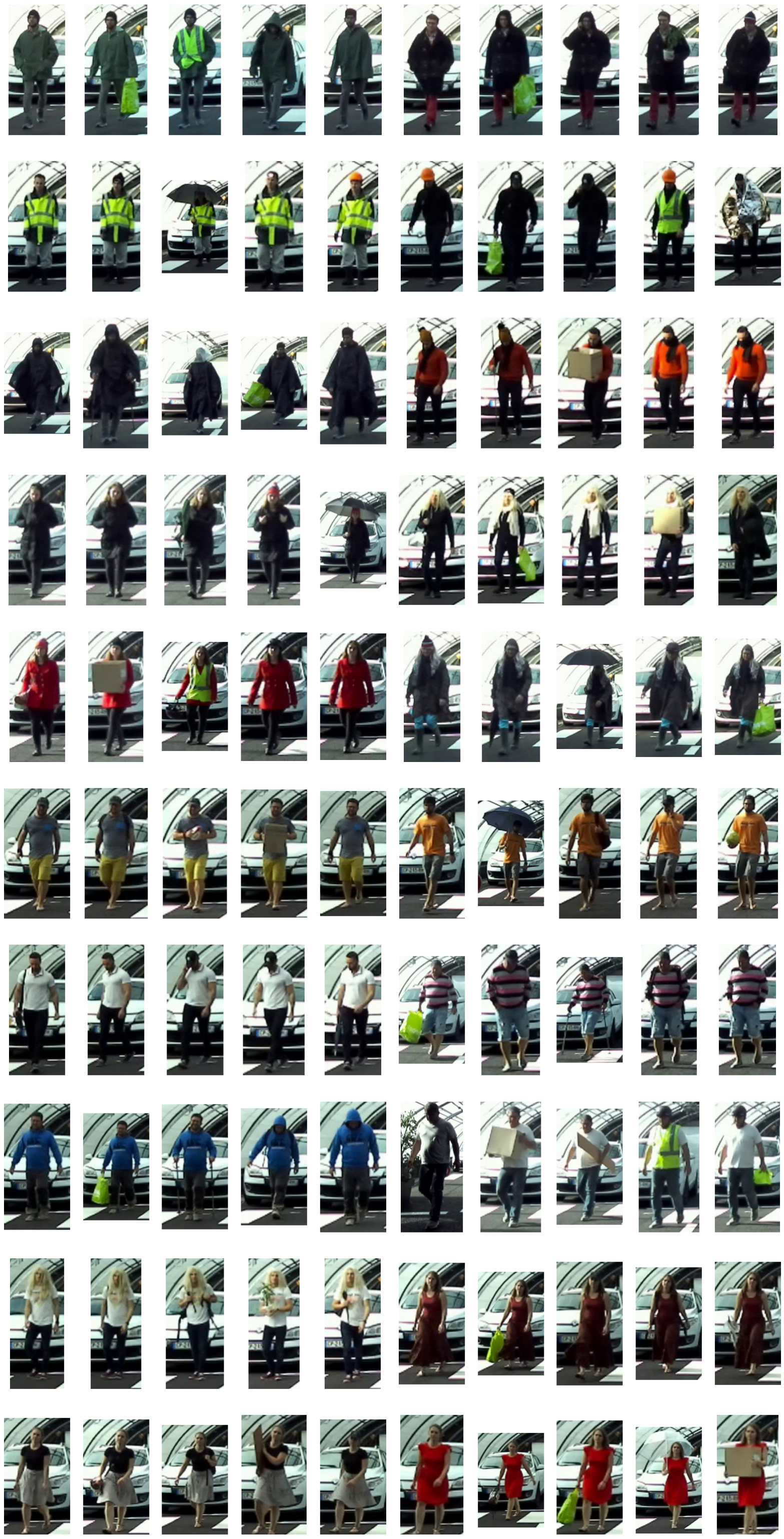


Figure 34: 100 Pedetrians of physical tests

2.2.2 Simulation tests

The simulation tests are performed by LNE using 4DVirtualiz (4DV), which is a digital twin software devoted to robotics and the automotive field. This simulator allows creating scenarios from scratch using the items included in the library of the software or by importing

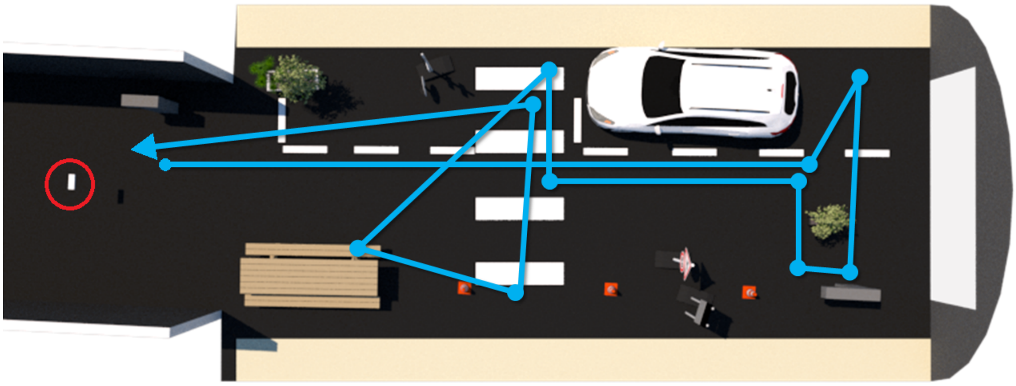


Figure 35: Human route.

our 3D models of building and vegetable ... etc. In this work, the 3D model of the PAVIN was produced by Cerema in SketchUp format. This model is then imported to Blender (see Figure 36) in order to generate fbx file which is required by 4DV simulator. The use of Blender is almost indispensable to generate the fbx file otherwise, some information may be lost.

Figure 36: PAVIN 3D model in Blender.

Once the fbx file has been imported into the 4DV simulator (see Figure 37), the scenario is configured by specifying the time of day (day or night), the weather conditions (clear or foggy), the humans (their appearance and route), and the cameras used to retrieve images. Two cameras are used here, the first being ZED2 to retrieve RGB images and a semantic camera to obtain the segmented images which are then used for the annotation step.

The time of day can be easily specified in the 4DV simulator by setting the simulator clock such as 12 pm for daytime and 8 pm for nighttime.

The weather conditions can also be specified in the 4DV simulator, however the visual rendering of the fog is very poor as shown in Figure 38. In fact, the fog intensity increases little with the distance and fog haze is practically non-existent. To overcome this problem, the smoke is used in addition to fog in order to enhance the visual rendering as shown in Figure 39. The smoke is set to zero speed, it does not move and there is no smoke ripple as shown in Figure.

Figure 37: PAVIN 3D model in 4DV.

In this study, two smoke intensities are defined to obtain weak and strong fog. In collaboration with Cerema, the smoke intensity values are set in such a way as to ensure visual acceptability.

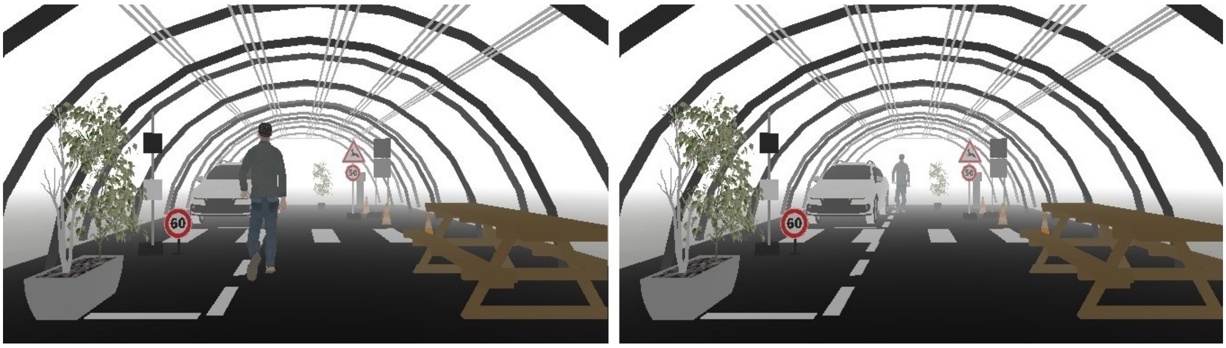


Figure 38: Original fog of 4DV.

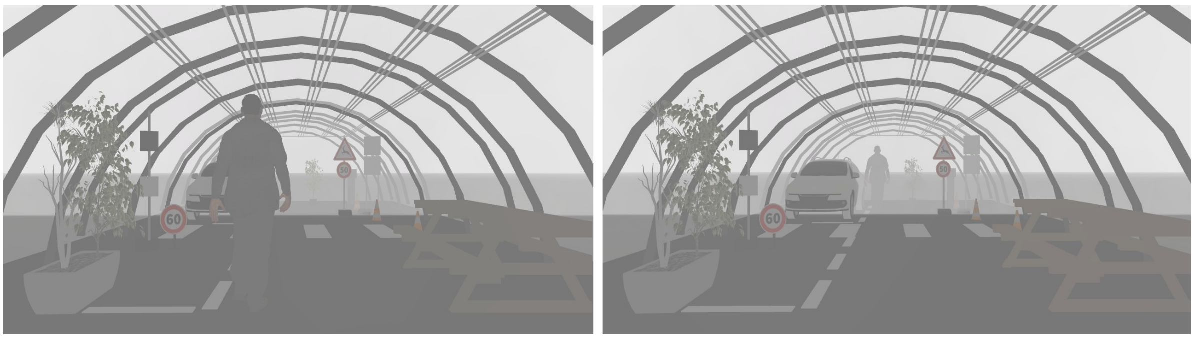


Figure 39: Fog using smoke of 4DV.

The 4DV library offer a wide choice of human in terms of gender, ethnicity, age and type of clothing. Figure 40 summarizes the humans used in the simulation. The appearance of the human can also be modified to change the hair or clothing color. In 4DV each human can be set with 3 appearances which means we have 36 different humans.

Figure 40: Library of 4DV.

The human route is defined to be as close as possible to the real route defined by Cerema, as shown in Figure 35.

4DV simulator includes ZED2i camera model which is close to that used by Cerema for test campaigns. Some of ZED2 parameters can be set as focus, zoom and frequency. The ZED2i is positioned at the same location as the real test, which is indicated in 3D model as shown in Figure 35 with the red circle. A semantic camera is also used to segment images retrieved by ZED2, facilitating the annotation process. The semantic camera is positioned as the same location as ZED2 and has the same zoom and frequency configuration as the ZED2.

4DV simulator offers automated test management, which means you specify the variable parameter in the scenario and let 4DV handle them automatically. Here, 3 variables for the weather conditions, 36 humans and 2 for time of day, which means 216 simulation tests to be executed by 4DV.

Concerning the annotation process, it is relatively easy and quick to use the 4DV annotation tool to generate a JSON file containing the bounding box of human detection for each single image.

2.2.3 Hybrid tests

The hybrid tests aims to narrow gap between simulation and physical tests by using the physical camera ZED2 instead of the simulated camera of 4DV. The physical camera is placed in front of a screen onto which images provided by 4DV are projected as shown in Figure 41.

The 4DV images are projected offline, not online, which means that 4DV does not run during the hybrid tests, but only the images retrieved from the simulation tests are projected onto the

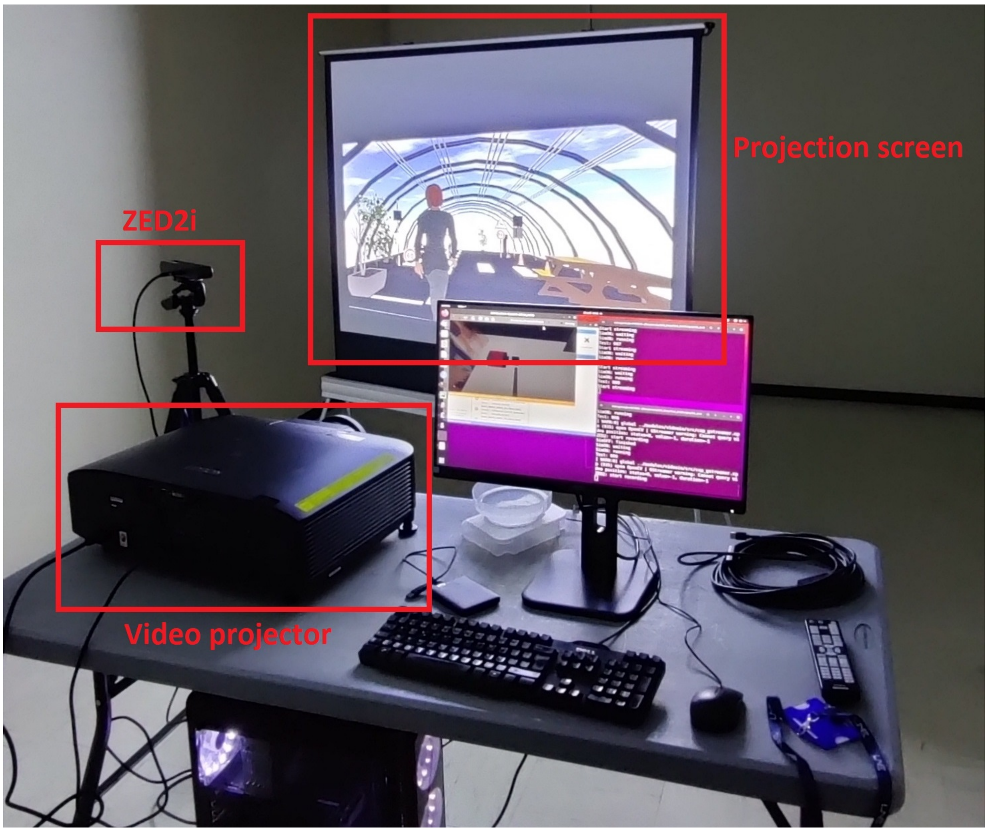


Figure 41: Darkroom of hybrid tests.

screen. To guarantee a high quality of retrieved images and avoid any light disturbance, the hybrid tests are carried out in a darkroom. The test are executed automatically and handled thanks to python scripts developed during this study. To illustrate, the following Figure 42 shows an image of hybrid test.

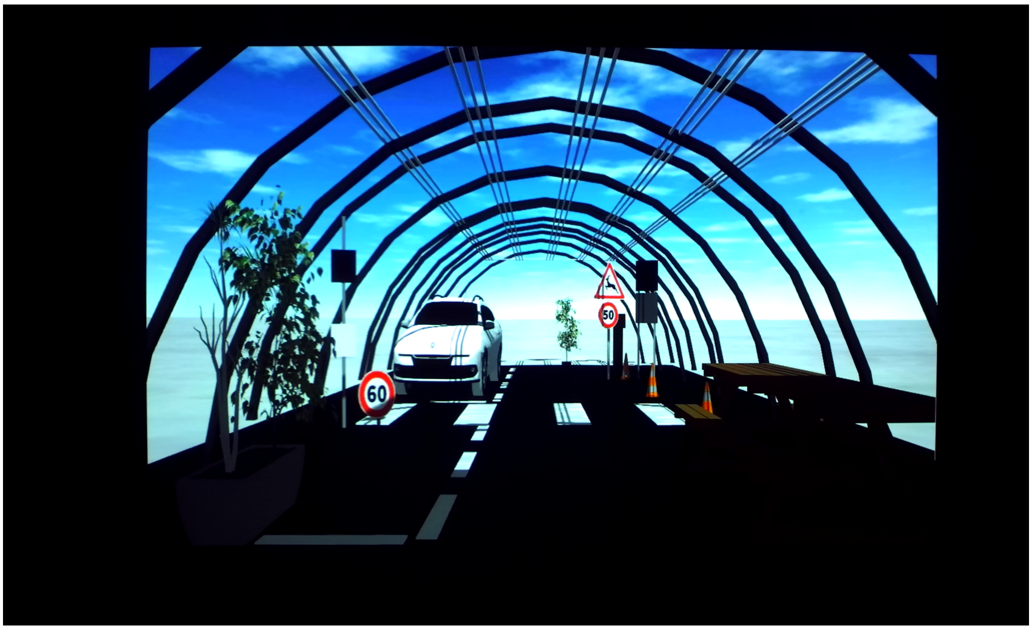


Figure 42: ZED2i image of hybrid test.

Once the hybrid tests have been finished, the obtained images are annotated based on JSON file generated by 4DV simulator during the simulation tests. Indeed, the bounding box of the image of the simulation test is projected on the image of the hybrid test to obtain the new bounding box.

As we can see in Figure 43, the bounding box of the 4DV image has as parameters ($B1$, B_h, B_w) and the idea is to compute the new parameters ($B1_{ZED}$, B_h_{ZED}, B_w_{ZED}) of the bounding box of the ZED image (see Figure 44. To do so, we can use the following expression:

$$\begin{aligned} h_{ratio} &= \frac{h_{ZED}}{h_{4DV}} \\ w_{ratio} &= \frac{w_{ZED}}{w_{4DV}} \\ B\_h_{ZED} &= B\_h * h_{ratio} \\ B\_w_{ZED} &= B\_w * w_{ratio} \end{aligned} \tag{1}$$

where $B1_{ZED} = [shift\_w + B1\_w * w_{ratio}, shift\_h + B1\_h * h_{ratio}]$ and $B1 = [B1\_w, B1\_h]$.

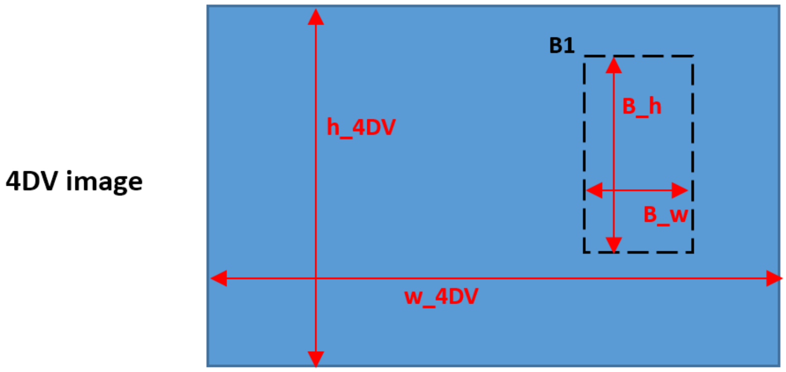


Figure 43: Bounding box on 4DV image.

As ZED2i camera retrieve the entire scene containing the 4DV image projection, including the black edge ($shift\_w$ and $shift\_h$), we need to remove this black edge from the ZED image to find the correct ZED image bounding box.

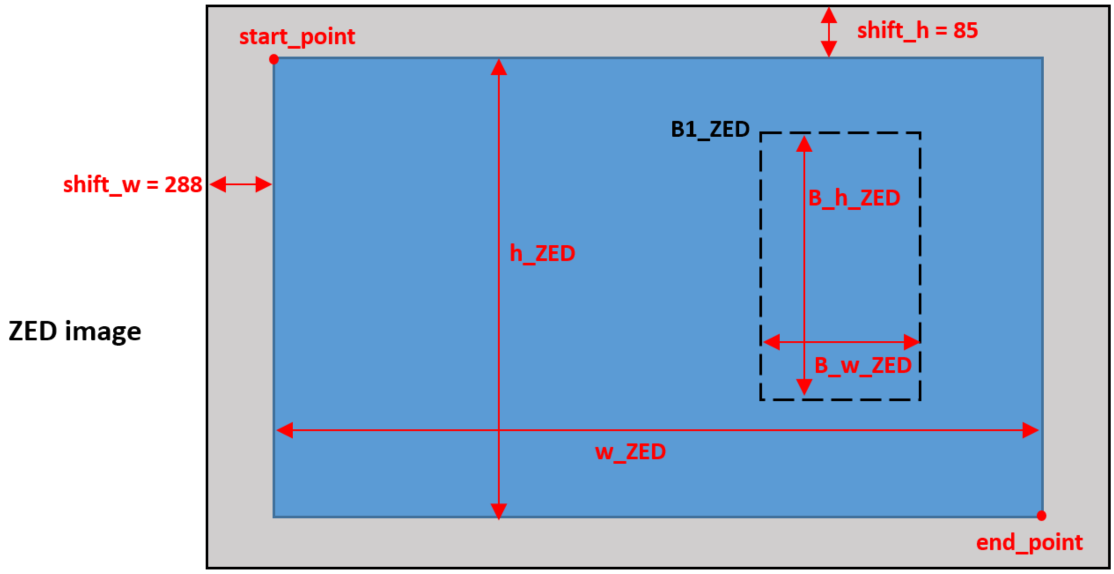


Figure 44: Bounding box on ZED image.

2.3 Method: A metric based on a pedestrian detection algorithm

As explained above, our approach to comparing and qualifying physical and digital test equipment is based on analysis of the results obtained by a detection algorithm, rather than on analysis of the raw images themselves. To do so, it is therefore necessary to have a pedestrian detection algorithm, a database labeled with a ground truth, and a detection algorithm evaluation metric. In this study, we have chosen to use the AUC score.

Concerning the detection algorithm, the third version of YOLO detection algorithm [10], which stands for "You Only Look Ones", was chosen in this analysis. It is indeed a very common algorithm in the literature on object detection. Moreover, it is very easy to handle. The library of objects available in this version contains 80 items. The algorithm requires two main parameters: the confidence threshold (a value between 0 and 1) of the labeling and the object to label in the images. Only the class "person" is labeled in this study and the confidence threshold chosen is explained in the following section. A frame can get multiple detections with different level of confidence even though only one pedestrian is walking in the scene into our database. As reminder, the objective is not the evaluation of YOLO algorithm but to use a popular object detection algorithm to evaluate main characteristics of the database, and to compare digital and physical artificial fogs.

In object detection, a metric widely used to evaluate the validity of a detection is the intersection over union (i.e. IOU) between bounding boxes as shown in Figure 45.

The intersection is calculated following the equation:

$$IOU(frame) = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

The precision-recall curve is then calculated based on the results of intersection over union values. The curve shows the trade-off between precision and recall for different confidence threshold values from the YOLO algorithm. As an example, the different detections obtained by the YOLO algorithm, for different levels of confidence, from 0.3 to 1, on two images from the database are presented on Figure 46. The left image of Figure 46 shows the 9 YOLO labels with two labels far from the pedestrian present in the scene, yet for one of them a confidence

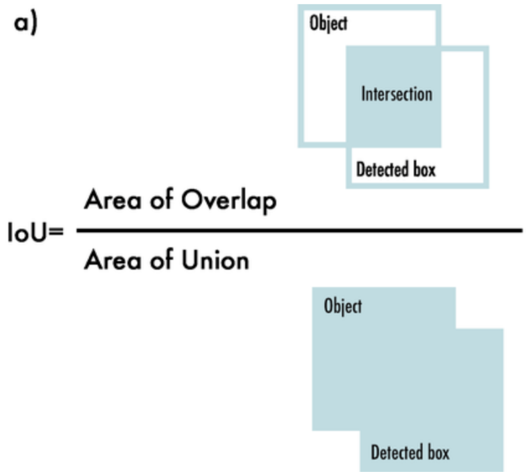


Figure 45: Intersection over Union (IoU). a) The IoU is calculated by dividing the intersection of the two boxes by the union of the boxes; b) examples of three different IoU values for different box locations [3].

value greater than 0.5. Then, the area under the curves (AUC) score is calculated. A large AUC value represents both high recall and high precision. A high precision value indicates a low false positive rate (good confidence value but no ground truth label), and a high recall value indicates a low false negative rate (low confidence value but ground truth has a label).

Figure 46: Example of YOLO detections on two Clear Weather images with different pedestrians. Colors : Green is for $confidence > 0.9$, Yellow is for $0.9 > confidence > 0.7$, Orange is for $0.7 > confidence > 0.5$, Red is for $0.5 > confidence > 0.3$.

2.4 Results and discussion

After labeling the daytime images using YOLO detection algorithm, the IOU has been calculated between the YOLO labeling and the ground truth labeling. In the literature, the IOU thresholds is often set at 0.5 for pedestrian and cyclists against 0.7 for vehicles [11]. From a relationship between the detection of true positives, false negatives and false positives we have calculated the precision and the recall and plotted the corresponding precision and recall curves (See Figure **??**) for an IOU of 0.5 and 0.7. The AUC has been calculated for each curve corresponding to the different sub-groups of the dataset (i.e. test type and weather condition), giving a score of YOLO labeling accuracy.

As we can see in Figure 47, only the simulation and the hybrid test are plotted, the analysis of the physical test data is in progress and will be given very soon. However, the scores obtained

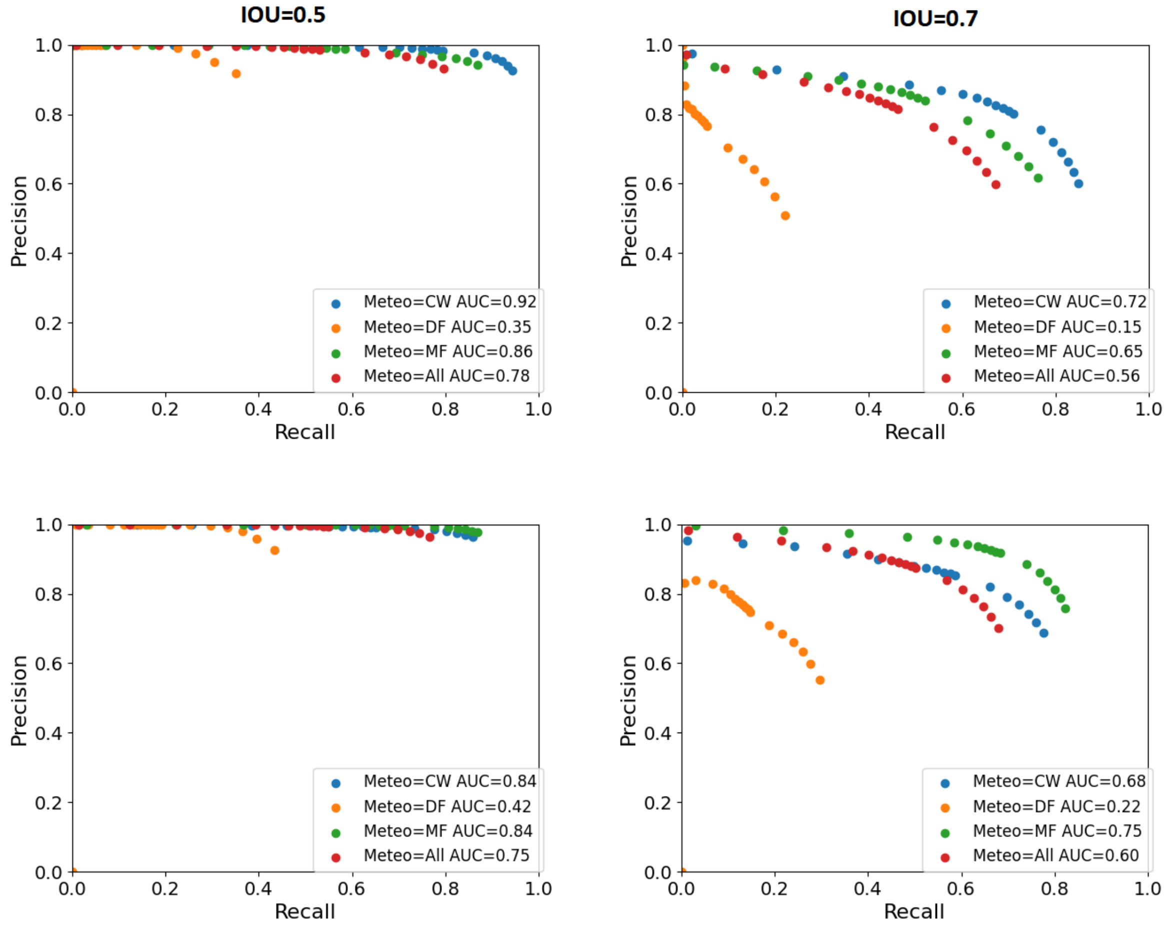


Figure 47: Precision and Recall curves with associated AUC values for the different sub-groups based on test type and weather condition with Simulation test (1st row), Hybrid test (2nd row), for an IOU of 0.5 (left column) and of 0.7 (right column).

for each sub-groups including physical test are given in Table 6. The scores of AUC on Table 6 for the IOU equal to 0.5 are stronger and closer to 1 than the one of IOU equal to 0.7, because we are less demanding in terms of detection.

It can been see that the precision and recall curves of Figure 47 corresponding to CF, MF and 'All weathers' have good results, with lower confidence values for MF than for CW. Concerning DF, the AUC are very low (below 0.3 for IOU $= 0.7$), whatever the type of test. In view of the results, YOLO is suitable for CW and MF conditions but is not adapted to the low visibility of DF condition from the tests. The results of the different tests remain very close, which allows us to say that YOLO is less sensitive to the source of the data, whether in the real or virtual world.

Test type	Weather condition	IOU$= 0.5$	IOU$= 0.7$
Physical	CW	0.93	0.74
	MF	0.89	0.64
	DF	0.29	0.08
Simulation	CW	0.92	0.72
	MF	0.86	0.65
	DF	0.35	0.15
Hybrid	CW	0.84	0.68
	MF	0.84	0.75
	DF	0.42	0.22

Table 6: AUC Scores of YOLO pedestrians detection depending on weather conditions and test type.

3. UTAC POC

3.1 Test program

3.1.1 Protocol version

The following scenarios refer mainly to the ENCAP 2023 protocol for the geometry, the corridors, and the test speeds. Then, some variants of known scenario have been created for this study.

3.1.2 Tests description

Refer to deliverable **L3.2** shared in January 2023.
The scenarios have been divided in 4 categories:

- Repeatability
  This category allows to evaluate the repeatability of the systems, it means to perform many times the same test, in the same conditions and check if the behavior is the same for all the repetitions.

- Robustness
  This category allows to evaluate the robustness of the systems, it means to perform a specific scenario and change different parameters (Speeds, Angles, visual aspect…) and see the behavior.

- Pre-critical
  This category allows to evaluate the anticipation of the systems on existing scenarios or new ones.

- Random
  This category allows to evaluate the systems in new random scenarios, unknown by the systems, and check the feasibility and the relevancy of it.

3.2 Vehicle under test

Three different vehicles have been tested during this campaign.

With all the vehicles, a first step of subjective testing has been done to have a first idea on the behavior of each vehicle.

After that we can select which tests are relevant to perform more precisely with measurement system. The tests performed and the results are detailed in 5.3.

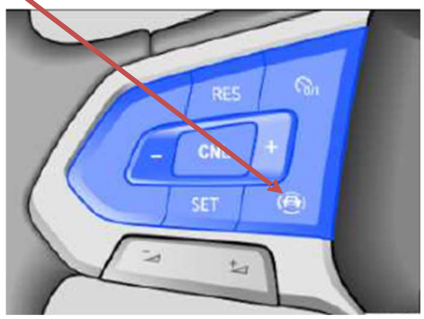
3.2.1 Description

<u>Vehicle n°1</u>: VW GOLF 8

The car is equipped with the ADAS system called “Travel Assist” allowing anticipating some situations.

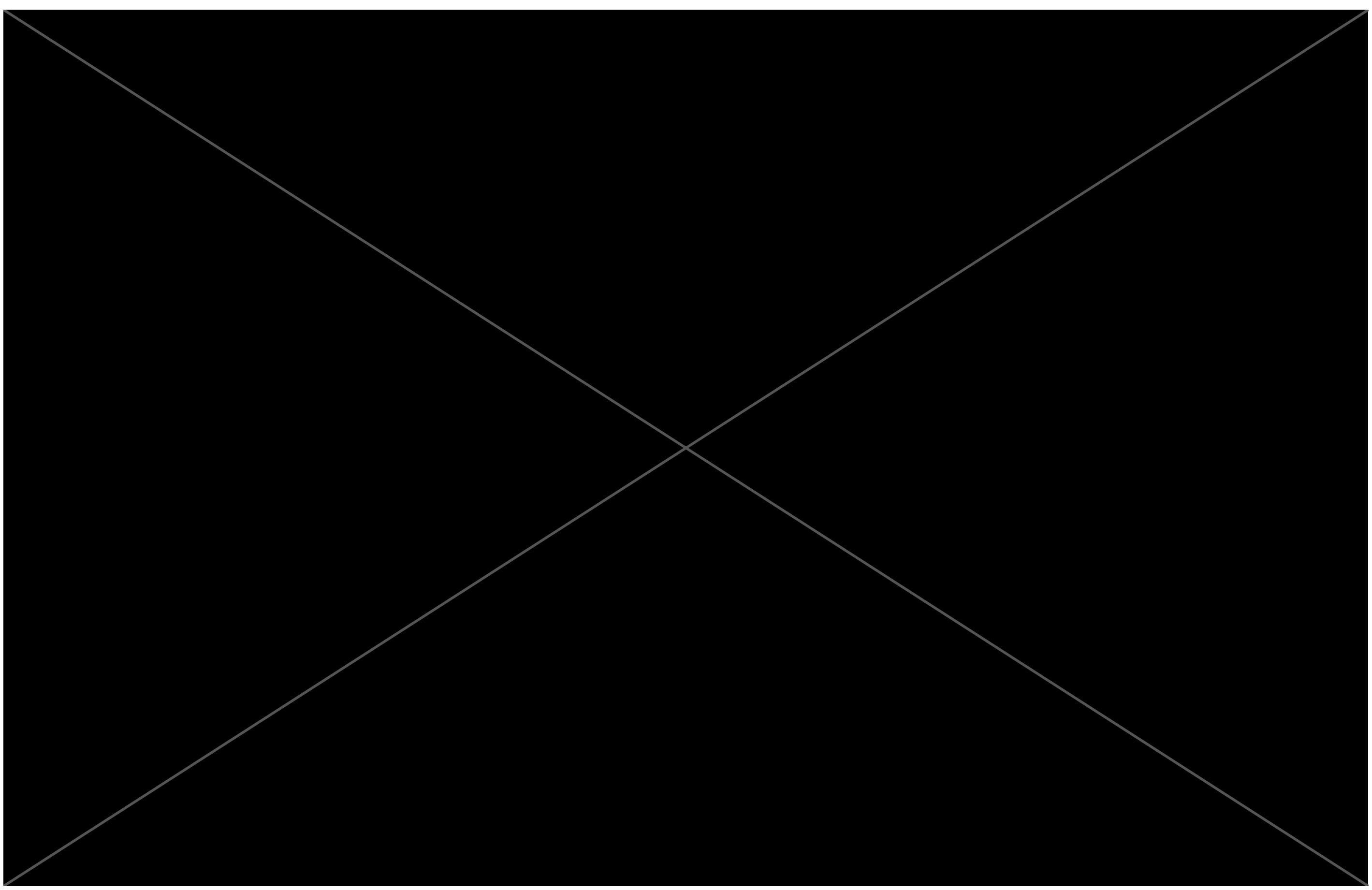
This system combines two driver assistance functions, Adaptive Cruise Control (ACC) for longitudinal assist and Lane Assist for lateral assist.

This function is activated by a button on the multifunction steering wheel, which therefore triggers longitudinal speed assist and lateral position assist.



For safety reasons, the driver must keep his hands on the steering wheel for the guidance to be effective.

To this longitudinal speed guidance can be added an anticipation function. The system calculates the position of the Golf based on GPS and route data from the navigation system and must adapt the speed in advance to the approach of bends, roundabouts, crossings, speed limit zones etc…
At the same time, it uses the traffic sign recognition system via the front camera and must adapt the speed as soon as a limitation is detected.



Vehicle n°3: ZOE prototype from French start-up NEXYAD with intelligent and anticipatory driving based on AI.

Two new innovative and intelligent proactive functions developed by NEXYAD:

Thus, NEXYAD has developed (and patented) two new functions of intelligent driving, and is in discussions with many French, German and Japanese manufacturers to market them: these driving functions use the estimation of the risk of NEXYAD and the consequently relevant safe speed to have to minimize the risk and stay in the green zone of driving risks (previous figure).

The risk is estimated according to the road map (arrival on a steep curve, a tight crossroads) and also what the vehicle sensors see (vehicle poorly parked, crowded crossroads, low field of vision

NEXYAD's 2 innovative and intelligent proactive driving functions are:

- **A safety assistant (named "safety coach") who alerts** the driver when his driving behavior is no longer prudent (risk to high) in relation to the driving context (accident reduction estimated by NEXYAD of at least 25%).

- **An intelligent and proactive ACC** that automatically regulates the vehicle speed according to the driving context (up to 75% accident reduction according to NEXYAD)

NB: The difference in the result between 25% reduction of accident in alert mode compared to 75% of the intelligent ACC mode is explained by the fact that the driver may not immediately and always consider the warnings and not slow down.

These two new functions are being implemented on a prototype vehicle, the NEXYAD DREAMOTOR1, **see photos below**, which is therefore one of the most advanced prototype in the world (On PRISSMA there is no French actor among vehicle manufacturers and it’s difficult to know their skills and developments on these very upstream and very competitive subjects). NEXYAD is part of the French industrial research and development group of the Regions Normandy/Ile de France, called NEXTMOVE (previously MOVEO), which supported and facilitated these innovative projects.

3.3 Testing equipment

3.3.1 Motion measurement

MOTION PACK 1	
Manufacturer Oxford Technical Solutions (OxTS)	
Unit model	
Sensors Accelerometers (Servo) / Gyros (MEMS)	
Data output rate 100 Hz	*Coupling method* GNSS / INS

3.3.2 Driving control system

CONTROLLER	
Manufacturer Antony Best Dynamics (ABD)	
Unit model XR Omni	
Sampling rate 100 Hz	
Analog input voltage ± 10 V	*A / D conversion* 16 bits

3.3.3 HMI analysis

VIDEO VBOX
Manufacturer Racelogic

GOPRO
Manufacturer GoPro

3.3.4 Additional equipment

POWERMESH ANTENNA		
	*Manufacturer* Antony Best Dynamics (ABD)	
	Unit model TrackFi PowerMesh	*Communication* Wifi 5 GHZ / 2.4 GHz
	UTAC unit reference	

BASE STATION		
	*Manufacturer* Oxford Technical Solutions	*Unit model* GPS-Base-2G GLONASS
	Correction format RTCM V3	*Position accuracy* < 2 cm
	UTAC unit reference	

Targets propulsion systems

GLOBAL VEHICLE TARGET PLATFORM

Manufacturer Anthony Best Dynamics (ABD)	
Platform unit model MKI / MKII P8500	*Communication* ABD Wifi 5 GHz
MP Unit model RT3002 / RT3002G	
UTAC unit reference DDV0228 / DDV0233 / DDV0273	

SOFT PEDESTRIAN TARGET RIG

Manufacturer Anthony Best Dynamics	*Unit model* SPT 20
Steering Robot SR60	*Sled height* 25 mm
Maximum speed 20 km/h with 15 kg payload	*Maximum acceleration* 0.8 g with 15 kg payload
UTAC unit reference	

EPT/EBT PLATFORMS

Manufacturer 4Active Systems (4A)	
Single belt unit model 4activeSB	*Dual belt unit model* 4activeSB
Single belt model dimensions Width : 492 mm Length: 990 mm Height: 26 mm Weight: 4kg	*Dual belt model dimensions* Width : 492 mm Length: 990 mm Height: 26 mm Weight: 4kg
Single UTAC unit reference -	*Dual UTAC unit reference* -

3.3.5 Road users targets

GUIDED SOFT TARGET

Manufacturer DRI
Unit model Hatchback Soft Car 360™ (Ford Fiesta)
Dimensions
UTAC unit reference

3.3.6 Vulnerable road users targets

EUROPEAN ADULT PEDESTRIAN TARGET

Manufacturer	4Active Systems (4A)
Unit model	4activePA-adult
Model dimensions	Body height: 1800 mm Shoulder width: 500 mm Weight: 4 kg
UTAC unit reference	HUM00XX

EUROPEAN CHILD PEDESTRIAN TARGET

Manufacturer	4Active Systems (4A)
Unit model	4activePA-child
Model dimensions	Body height: 1154 mm Shoulder width: 298 mm Weight: 2 kg
UTAC unit reference	HUM00XX

EUROPEAN BICYCLIST TARGET

Manufacturer	4Active Systems (4A)
Bicyclist unit model 4activeBS-adult	*Bike unit model* 4activeBS-adult
Bicyclist model dimensions Body height: 1800 mm Shoulder width: 500 mm Weight: 4 kg	*Bike model dimensions* Handlebar height : 1200 mm Wheelbase: 1230 mm Weight: 6 kg
Bicyclist UTAC unit reference HUM00XX	*Bike UTAC unit reference* -

3.3.7 Road signs targets

SPEED LIMIT 50 / 70 / 90

Speed limit type	B14 - Explicit speed 50 - 70 - 90
Dimensions Diameter 1050 mm	*Specification* Class 2
UTAC unit reference	-

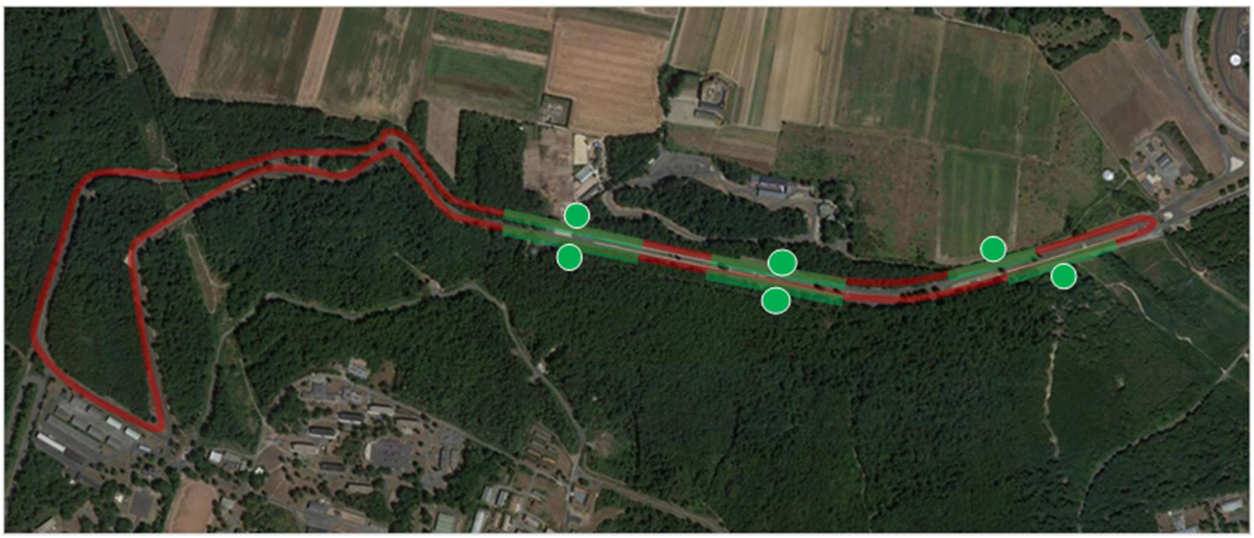
3.4 UTAC test tracks

3.4.1 Location

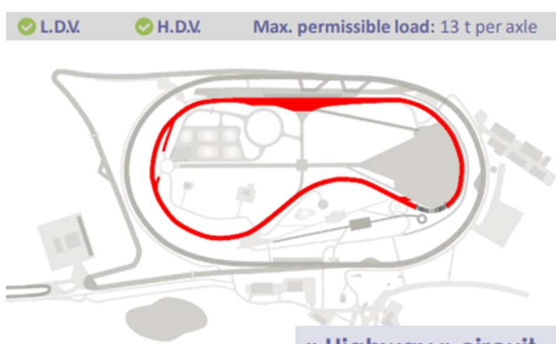
MONTLHERY TEST UNIT	
	*Location* Autodrome de Linas-Montlhéry *Address* BP20212 91311 Montlhéry CEDEX France

3.4.2 Specifications

3.4.2.1 Montlhéry Unit – CR



3.4.2.2 Montlhéry Unit – TEQMO Highway

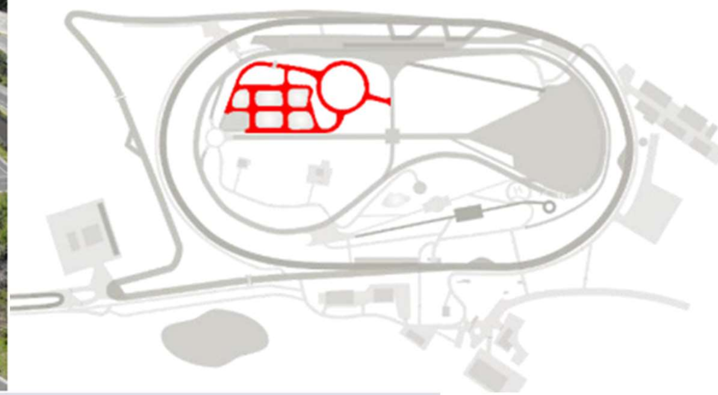


« Highway » circuit

Features
- Length: 2.2 km
- Width: 3.5 m
- 2 and 3 lanes with multiple entries & exits
- Tunnel and tollbooth
- Road marking & signaling

Services
- Platooning
- Adaptive cruise control
- Lane keeping
- Road sign recognition

3.4.2.3 Montlhéry Unit – TEQMO City



« City » zone

Features

- Area : 38,500 m²
- 2 and 3 lanes 3.5 m wide
- Lighting & signaling
- 3 crossroads incl. 1 with roundabout and 1 with traffic lights, 7 « T » intersections
- 1 roundabout 50 m radius, 1 level crossing

Services

- Autonomous Shuttle
- Various projects
- Bus stations

3.5 Testing results

3.5.1 Post-processing

We define the PASS/FAIL as:
- PASS: The system reacted and allowed to avoid the collision
- FAIL: The system didn't react OR reacted to late to avoid the collision

To go further in the analysis, we check the following values in the raw data (.txt file):
- **Maximum Speed (kph)** of the vehicle during the test

  For that, we use the channel named "Speed (kph)" and we check the maximum during the test.

- **Minimum distance (m)** between the vehicle and the Target

  This distance is 0 in case of Impact and in case of avoidance we use the channels named "Speed (kph)" and "Relative Longitudinal Distance (m)".
  First, we find the index where the vehicle stops, it means when "Speed (kph)" reaches 0 kph.
  Then, we check the "Relative Longitudinal Distance (m)" value at the same index.

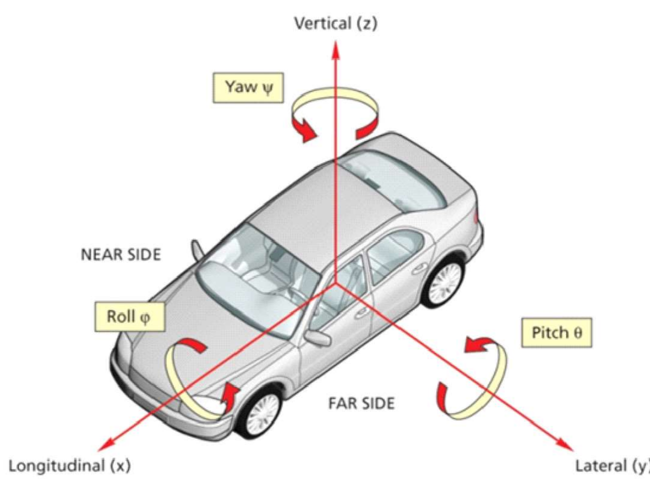
- **Vehicle Impact Speed (kph)** in case of impact

  This is the Vehicle Speed at the time of collision with the Target. We use the channels named "Speed (kph)" and "Relative Longitudinal Distance (m)".
  First, we find the index of the collision, it means where "Relative Longitudinal Distance (m)" reaches 0 m.
  Then, we check the "Speed (kph)" value at the same index.

- **Vehicle Speed (kph)** at driver avoidance in case of it.

  This is the Vehicle Speed at the time of driver avoidance (steering or braking). Depending on the action, we can find the index of the avoidance (huge variation) using "Yaw Velocity (°/s)" or "Forward Acceleration (m/s²)".
  Then we check the "Speed (kph)" value at the same index.

3.5.2 Reference data system



3.5.3 Details of tests performed and result table

Vehicle n°1: VW GOLF 8

a) **Repeatability:**

First, three scenarios from ENCAP have been tested without any measurement equipment, to check which can be relevant or not:

Category	Scenarios	Number of subjective tests	Successful	Keep for objective tests
Repeatability	CPNCO	2	NO	NO
	CPFA	3	NO	NO
	CBLA	3	YES	YES

The two crossing scenarios (CPNCO and CPFA) are not relevant for this car, contrarily to the longitudinal one (CBLA) which has been selected for objective testing.
Then, 10 repetition of the same scenario CBLA have been performed and all the tests were successful (PASS).

Category	Scenarios	Number of objective tests	Successful
Repeatability	CBLA	10	YES

Here is the post-processing of the Repeatability part:

Scenario	Date	Time	Nbr of test	VUT Speed (kph)	Overlap	Success	VUT reaction	Anticipation	Comments	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
CBLA	16/02/2023	10:00	1	30	50%	YES	YES	YES	ACC REGULATE, OVERLAP CLOSE TO 25%	28,35	5,3	0	0
CBLA	16/02/2023	10:05	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	32,21	3,93	0	0
CBLA	16/02/2023	10:11	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	31,55	4,06	0	0
CBLA	16/02/2023	10:15	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	29,15	4,48	0	0
CBLA	16/02/2023	10:19	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	29,06	4,46	0	0
CBLA	16/02/2023	10:22	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,02	4,79	0	0
CBLA	16/02/2023	10:25	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,3	4,82	0	0
CBLA	16/02/2023	10:28	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	32,36	3,6	0	0
CBLA	16/02/2023	10:32	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	30,46	4,14	0	0
CBLA	16/02/2023	10:35	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,09	4,99	0	0
CBLA	16/02/2023	10:39	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,23	5,06	0	0

b) Robustness:

In the same way, we started to perform the scenarios without equipment to see the relevancy:

Category	**Scenarios**	**Number of subjective tests**	**Successful**	**Keep for objective tests**
Robustness	**CBLA**	**1**	**YES**	**YES**
	Stationary CAR on emergency lane (new scenario)	**1**	**YES**	**YES**
	Stationary object on highway	**2**	**NO**	**NO**

The last scenario with an object on the road is not relevant with this car. The CBLA and the Stationary CAR are relevant; we selected those 2 for the next step.

Then, each scenario has been performed 10 times by changing different parameters (like Objects Speed, Angles, Overlaps…):

Category	**Scenarios**	**Number of objective tests**	**Successful**
Robustness	**CBLA**	**10**	**YES**
	Stationary CAR on emergency lane	**10**	**YES**

All the tests are successful, here are the details of the post-processing:

Scenario	Date	Time	Nbr of tests	ACC/AEB	VUT Speed (kph)	Overlap	Target Speed (kph)	camera	wipers with liquid	dummy accessories	Objec in the field of view	Success	Reaction	Anticipation	Comment	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
CBLA	16/02/2023		1	AEB	30	50%	15	clean	NO	standard	NO	NO	YES	NO	AEB LATE, DRIVER BRAKE	29,42	0	0	27,61
CBLA	16/02/2023	12:20	1	ACC	30	50%	15	dirty	NO	standard	NO	YES	YES	YES	acc regulate	30,79	4,02	0	0
CBLA	16/02/2023	11:50	1	ACC	30	50%	15	clean	NO	Yellow jacket	NO	YES	YES	YES	acc regulate	28,22	4,91	0	0
CBLA	16/02/2023	11:56	1	ACC	30	50%	15	clean	NO	jacket + backpack	NO	YES	YES	YES	acc regulate	30,01	4,25	0	0
CBLA	16/02/2023	12:05	1	ACC	30	50%	15	clean	YES	standard	NO	YES	YES	YES	acc regulate	29,15	4,46	0	0
CBLA	16/02/2023	11:35	1	ACC	30	75%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,11	4,96	0	0
CBLA	16/02/2023	11:19	1	ACC	30	25%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,16	4,91	0	0
CBLA	16/02/2023	11:40	1	ACC	30	0%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	27,89	5,24	0	0
CBLA	16/02/2023	11:45	1	ACC	30	100%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,23	1,53	0	0
CBLA	16/02/2023	12:16	1	ACC	30	50%	15	clean	NO	standard	parked car	YES	YES	YES	acc regulate	27,78	4,3	0	0

Robustness is not 100% perfect, but 90% with 1 impacts (or test driver manuel avoidance) among 10 tests, as shown in red in the above table.

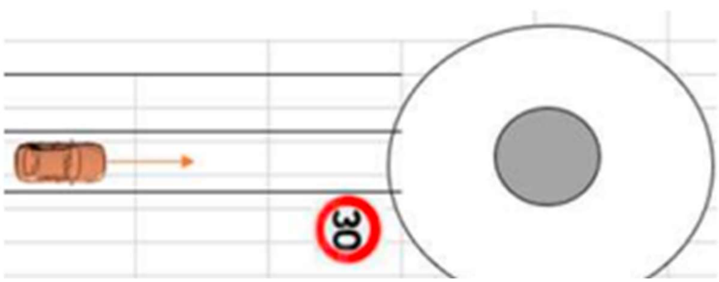
Scenario	Date	Time	Nbr of tests	ACC/AEB	VUT Speed (kph)	Overlap	Angles of driving (°)	objet	Target with roofbox	Success	Reaction	Anticipation	Comment	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
stationnary car on highway	16/02/2023	15:10	1	ACC	30	100%	0	NO	NO	YES	YES	YES		29,02	2,61	0	0
stationnary car on highway	16/02/2023	15:13	1	ACC	40	100%	0	NO	NO	YES	YES	YES		38,72	5,33	0	0
stationnary car on highway	16/02/2023	15:16	1	ACC	50	100%	0	NO	NO	YES	YES	YES		47,34	4,13	0	0
stationnary car on highway	16/02/2023	15:24	1	ACC	30	50%	0	NO	NO	NO	YES	NO	fcw very late and AEB when drive avoid (steering)	28,98	2,26	0	24,18
stationnary car on highway	16/02/2023	15:30	1	ACC	30	75%	0	NO	NO	YES	YES	YES		28,74	3,3	0	0
stationnary car on highway	16/02/2023	15:33	1	ACC	30	-75%	0	NO	NO	YES	YES	YES		29,01	3,55	0	0
stationnary car on highway	16/02/2023	15:40	1	ACC	30	100%	5	NO	NO	YES	YES	YES		29,08	3,55	0	0
stationnary car on highway	16/02/2023	16:12	1	ACC	30	100%	22	NO	NO	NO	NO	NO	no reaction (steering)	29,27	1,79	0	28,83
stationnary car on highway	16/02/2023	16:00	1	ACC	30	100%	0	YES	NO	YES	YES	YES		28,87	2,72	0	0
stationnary car on highway	16/02/2023	15:50	1	ACC	30	100%	0	NO	YES	YES	YES	YES		29,28	2,97	0	0

As for any vehicle, robustness is not 100% perfect, but 80% with 2 impacts (or test driver manual avoidance) among 10 tests, as shown in red in the above table.

c) Pre-critical:

First, we tested the vehicle without equipment some situation that can generate anticipation of the system:

Category	Scenarios	Number of subjective tests	Successful	Keep for objective tests
Pre-critical	Improperly parked vehicle	1	YES	YES
	Approach to roundabout	1	NO	NO
	Close and misleading traffic sign	1	YES	YES

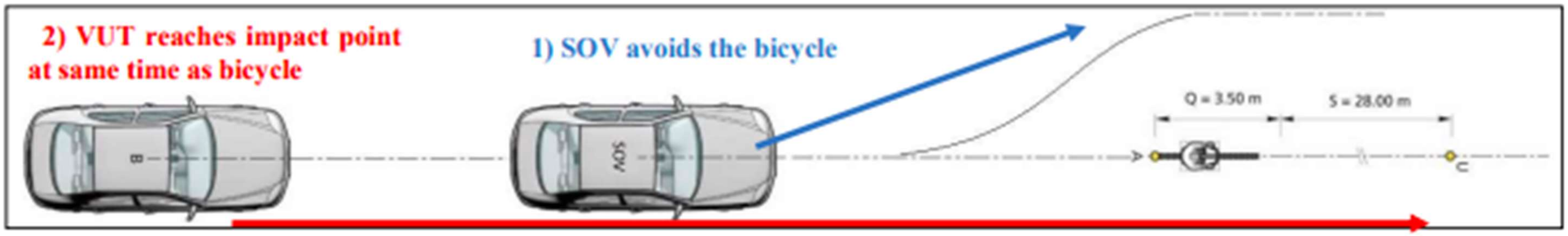


Then we performed some situation with measurement equipment.
All the tests are successful; here are the details of the post-processing:

Scenario	Date	Time	Nbr of tests	VUT Speed (kph)	Success	Reaction	Anticipation	Comment	max speed before traffic sign (kph)	speed after 50kph traffic sign (kph)
Highway driving (close and misleading traffic sign 50kph EXIT)	08/03/2023	11:07	1	ACC regulation	NOK	ACC	ACC	Detection of the 50kph traffic sign (for EXIT) and speed adaptation (false positive)	85,26	48,97
Highway driving (close and misleading traffic sign 50kph EXIT)	08/03/2023	11:13	1	ACC regulation	NOK	ACC	ACC	Detection of the 50kph traffic sign (for EXIT) and speed adaptation (false positive)	85,74	48,46

d) <u>Random :</u>

We performed a new scenario that we imagined for this campaign; this is a CPLA merged with a Cut-Out:



For this scenario, the Golf 8 had a good reaction, the VUT first regulates its speed to keep a safe distance with the SOV, and then after the SOV performed its Cut-out, the VUT regulates behind the bicycle.

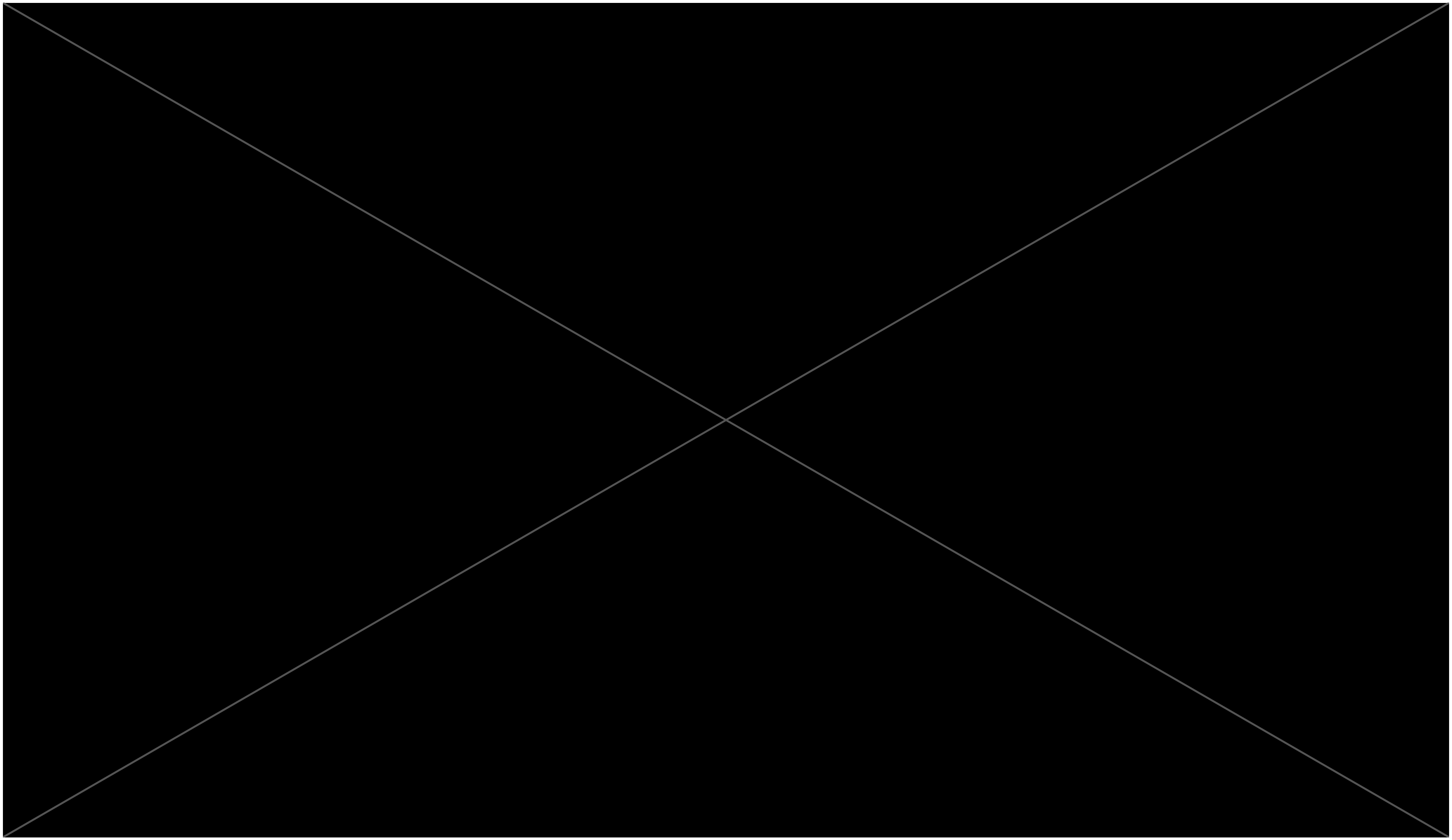
This test is relevant, feasible and interesting to propose for future studies.

Scenario	Date	Nbr of test	VUT Speed (kph)	Overlap	Success	VUT reaction	Anticipation	Comments	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
Longitudinal Bicyclist with VUT preceded by a vehicle	17-févr	1	70	50%	YES	YES	YES	ACC regulation on SOV then bike detection, then ACC regulation on bike	51,78	120,81	0	0

[redacted]

[redacted]

[redacted]



[redacted]

Vehicle n°3 Zoe NEXYAD:

a) **Repeatability:**

First, three scenarios from ENCAP have been tested without any measurement equipment, to check which can be relevant or not:

Category	Scenarios	Number of subjective tests	Successful	Keep for objective tests
Repeatability	CPNCO	3	NO	NO
	CPFA	2	YES	YES
	CBLA	3	YES	NO

The two scenarios CPFA and CBLA are relevant, we decided to keep the CPFA which is more challenging according to VALEO.
Then, only 5 repetitions of the same scenario CPFA have been performed , because the NEXYAD prototype supports different new functionalities and was not fully dedicated nor optimized for these tests, and 80% of the tests were successful (PASS).

Category	Scenarios	Number of objective tests	Successful
Repeatability	CPFA	5	4 PASS and 1 FAIL

Here is the post-processing of the Repeatability part:

Scenario	Date	Nbr of Tests	VUT Speed (kph)	Target Speed (kph)	Success	Reaction	Anticipation	Comment	max speed (kph)	remaining distance (m)	aviodance speed (kph)	impact speed (kph)
CPFA	05/07/2023	1	ACC	3	Yes	Yes	Yes	Vehicle reaction	38,02	1,33	0	0
CPFA	05/07/2023	1	ACC	3	No	No	No	Driver braking	32,04	0,28	27,87	0
CPFA	05/07/2023	1	ACC	3	Yes	Yes	Yes	Vehicle reaction	33,53	0,42	0	0
CPFA	05/07/2023	1	ACC	3	Yes	Yes	Yes	Vehicle reaction	29,54	1,93	0	0
CPFA	05/07/2023	1	ACC	3	Yes	Yes	Yes	Vehicle reaction	31,13	14,01	0	0

As for any vehicle, repeatability is not 100% perfect, but 80% with 1 impacts (or test driver avoidance) among 5 tests , as shown in red in the above table.

b) Pre-critical:

First, we tested the vehicle without equipment some situation, which can generate anticipation of the system:

Category	Scenarios	Number of subjective tests	Successful	Keep for objective tests
Pre-critical	Stationary car	1	NO	NO
	Approach to roundabout	1	YES	NO
	Approching strong curve	1	YES	YES

Then we performed some situation with measurement equipment.

All the tests are successful, here are the details of the post-processing:

Scenario	Date	Nbr of tests	VUT Speed (kph)	Success	Reaction	Anticipation	max speed (kph)	speed after curve 1 (kph)	speed after curve 2 (kph)
Approching Curve	03/07/2023	1	ACC	yes	yes	yes	83	29,9	53,9
Approching Curve	03/07/2023	1	ACC	yes	yes	yes	84	29,5	50,7

[redacted]

3.7. Annex

Data channels definitions

TIME INFORMATIONS		
Channel names	*Units*	*Comments*
Time	s	Time starts in the path
MP Time	s	GPS time of VUT
MP Time Tracker 1	s	GPS time of VRU or GST

VUT SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Actual X (front axle)	m	X of the car (VUT) (at the bumper)
Actual Y (front axle)	m	Y of the car (VUT) (at the bumper)
Speed	kph	Absolute speed of the car (VUT)
Forward velocity	m/s	Forward speed of the car (VUT)
Lateral velocity	m/s	Lateral speed of the car (VUT)
Forward acceleration	m/s^2	Forward acceleration of the car (VUT)
Lateral acceleration	m/s^2	Lateral acceleration of the car (VUT)
Yaw angle	°	Yaw angle of the car (VUT)
Yaw velocity	°/s	Yaw velocity of the car (VUT)
Yaw acceleration	$°/s^2$	Yaw acceleration of the car (VUT)

TARGET SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Head tracker reference X position	m	Position of the VRU on X axis
Head tracker reference Y position	m	Position of the VRU on Y axis
Head tracker forward velocity	m/s	Speed of the VRU on its path
Head tracker forward acceleration	m/s^2	Acceleration of the VRU on its path

RELATIVES VUT/TARGET SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Time to Collision (longitudinal)	s	Remaining time before the VUT strikes the target, assuming that the VUT and the target would continue to travel with the speed it is travelling
Relative longitudinal distance	m	Difference between the longitudinal positions of the vehicle and the target
Relative lateral distance	m	Difference between the lateral positions of the vehicle and the target
Relative longitudinal velocity	m/s	Difference between the longitudinal speeds of the vehicle and the target
Relative lateral velocity	m/s	Difference between the lateral speeds of the vehicle and the target
Relative yaw	°	Difference between the yaw angles of the vehicle and the target

REFERENCES

[1] IFSTTAR, "Transpolis: A shared innovation platform," 2021. [Online]. Available: https://www.ifsttar.fr/en/research-expertise/major-projects/partnership-projects/transpolis/

[2] Transpolis, "Transpolis: Les fromentaux test centre," 2021. [Online]. Available: https://transpolis.fr/les-fromentaux

[3] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[4] T. Genevois, J.-B. Horel, A. Renzaglia, and C. Laugier, "Augmented reality on lidar data: Going beyond vehicle-in-the-loop for automotive software validation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 971–976.

[5] L. Rummelhard, A. Negre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2485–2490.

[6] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[7] T. Genevois, L. Rummelhard, A. Spalanzani, and C. Laugier, "A predictive collision detector using probabilistic occupancy grids for a global approach to collision avoidance," under review in IEEE International Conference on Robotics and Automation (ICRA), 2023.

[8] J.-B. Horel, R. Baruffa, L. Rummelhard, A. Renzaglia, and C. Laugier, "A navigation-based evaluation metric for probabilistic occupancy grids: Pathfinding cost mean square error," in the proceedings of 26th IEEE International Conference on Intelligent Transportation Systems (ITSC), 2023.

[9] M. Colomb, K. Hirech, P. André, J. Boreux, P. Lacôte, and J. Dufour, "An innovative artificial fog production device improved in the european project "fog"," *Atmospheric Research*, vol. 87, no. 3, pp. 242–251, 2008, third International Conference on Fog, Fog Collection and Dew. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169809507002037

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[11] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-yolo: Real-time 3d object detection on point clouds," *arXiv preprint arXiv:1803.06199*, 2018.