ReliablE in-Vehicle pErception and decisioN-making in complex environmenTal conditionS

Grant Agreement Number: 101069614

D.3.2: Perception System and Self-Assessment

Document Identification			
Status	Final	Due Date	31-08-2024
Version	1.0	Submission Date	27-12-2024
Related WP	WP3	Document Reference	D3.2
Related Deliverable(s)	D3.1	Dissemination Level	PU
Lead Participant	TUD	Document Type:	OTHER
Contributors	All WP3 partners	Lead Authors	Dariu Gavrila (TUD)
		Reviewers	Kostas Koufos (WMG)
			Michael Buchholz (UULM)

Funded by the European Union

This project has received funding under grant agreement No 101069614. It is funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Document Information

Author(s)		
First Name	**Last Name**	**Partner**
Kirsty	Aquilina	APTIV
Piotr	Kocanda	APTIV
Alireza	Ahrabian	HIT-UK
Nikolaos	Toulios	HIT-UK
Quan	Nguyen	HIT-FR
Anthony	Ohazulike	HIT-FR
Massimiliano	Lenardi	HIT-FR
Anastasia	Bolovinou	ICCS
Markos	Antonopoulos	ICCS
Giorgos	Hatzipavlis	ICCS
Andras	Palffy	PERCIV
Leonardo	González Alarcón	TECN
Javier	Araluce	TECN
Alberto	Justo	TECN
Dariu	Gavrila	TUD
Ted	De Vries Lentsch	TUD
Thomas	Griebel	UULM
Mehrdad	Dianati	WMG
Hakan	Yatbaz	WMG
Sajjad	Mozaffari	WMG

Document History			
Version	**Date**	**Modified by**	**Modification reason**
0.1	11/11/2024	TUD	First overall Deliverable draft based on partner input. Sections on T3.1, EXP1 and Conclusions still missing. Cross-referencing of citations and figures still to be done.
0.2	27/11/2024	TUD	First overall Deliverable draft based on partner input. Sections on T3.1, EXP1 and Conclusions added. Cross-referencing of figures/tables/citations done.

Document History			
Version	**Date**	**Modified by**	**Modification reason**
0.3	04/12/2024	TUD	New document version after partner feedback and review from WMG.
0.4	23/12/2024	TUD	New document version after partner feedback and review from UULM
1.0	27/12/2024	ICCS	Final review by the project's coordinator and submission in the EC portal

Quality Control		
Role	**Who (Partner short name)**	**Approval Date**
Deliverable leader	Dariu Gavrila (TUD)	23/12/2024
Quality manager	Panagiotis Lytrivis (ICCS)	27/12/2024
Project Coordinator	Angelos Amditis (ICCS)	27/12/2024

Executive Summary

Work Package (WP) 3 addresses the development of the environment perception system and its self-assessment (SA) within the EVENTS project. The environment perception system involves on-board sensing (using camera, radar, LiDAR), supported by localization technology (GNSS+INS) and High-Definition (HD) digital maps, and is potentially augmented by Vehicle-to-Everything (V2X) communication technologies. WP3 provides the algorithmic content of the perception modules described in EVENTS architecture (see Deliverable (D) 2.2 **Error! Reference source not found.**) to address t he set of challenging driving scenarios, called Experiments (EXP1-EXP8), specified in D2.1 **Error! Reference source not found.**.

D3.2 covers all work done within WP3 and its tasks (T3.1-T3.5). T3.1 involves the acquisition and adaptation of training data needed for the machine learning-based approaches. T3.2 covers the topic of semantic scene analysis and precise localization. T3.3 involves work on the integration of past and current measurements from on-board sensors to obtain the current environment state. Furthermore, it involves a prediction of how the latter will evolve over time. T3.4 is on the topic of augmented perception by V2X, extending the on-board perception of the ego-vehicle with information coming from other Connected and Automated Vehicles (CAVs) or infrastructure sensors. Finally, T3.5 covers perception system SA.

Rather than structuring D3.2 by the beforementioned tasks, it is structured by the Experiments (EXP1 – EXP8). This provides a more integrated view of how the various tasks work together in the perception subsystem to address the different driving scenarios. The sole exception is **T3.1**, which is discussed separately, as it pertains to training data acquisition and adaptation that in principle applies to multiple Experiments. Specifically, in T3.1, we explore the usage of existing public datasets, describe a newly acquired road debris dataset, cover data generation based on manual annotation and simulation, and present an approach for self-supervised learning for object detection.

EXP1 addresses safe, comfortable, and time-efficient automated driving in complex urban environment while interacting with VRUs (e.g. pedestrians, cyclists). A LiDAR-based environment perception pipeline was developed, combining an object detector trained on a set of predefined classes (e.g. car, bike, and pedestrian) with a class-independent obstacle segmentation. A multi-object tracker tracks detected objects across time, and a motion prediction component predicts the future path of each tracked object. **EXP2** deals with the reconfiguration of a platoon formation after a split due to a roundabout. A cooperative motion prediction framework was developed based on a state-of-the-art (SOTA) model. The results demonstrate that V2V-enhanced predictions achieve a better understanding of the traffic scene. **EXP3**

concerns with safe automated driving in a complex urban environment with occlusion. It demonstrates the integration of reliability assessment outputs of environment state estimation (on-board SA methods) and V2X data into an onboard perception system. The main outcome of EXP3 is the development of an SA approach for object-tracking algorithms. **EXP4** addresses decision making for motion planning when faced with roadworks, unmarked lanes and narrow roads with assistance from perception SA. A pipeline for updating a pre-existing HD map under road work conditions was developed. The model assumes traffic bollards are being used to separate drivable vs non-drivable lanes, where such bollards are then used to determine the updated lane boundary. **EXP5** involves predictive perception and perception SA at merger onto the highway. A predictive perception system was developed that can reliably detect and track multiple 3D objects moving at various speeds in real-time and forecast their future movements based on historical trajectories. Perception monitoring through consistency checking is also implemented. **EXP6** concerns the sensing of small objects and semantic representation of these objects (relative position, height, object velocity, over-drivability and estimation of time to collision) within diverse weather conditions. An over-drivability classifier has been trained on the newly collected debris dataset with promising results. **EXP7** considers localization and perception SA mechanisms for advanced ACC under adverse weather or adverse road conditions. SA mechanisms for LiDAR-based 3D object detection and relative localization to the leading vehicle were developed. These mechanisms were evaluated using public datasets and demonstrated superior performance compared to the current state-of-the-art. **EXP8** concerns emergency evasion maneuver under adverse weather conditions including perception SA. A radar point cloud segmentation network was developed to provide object detection, ego-motion estimation, and SA as an input for the maneuver planning.

Table of Contents

List of Tables

List of Figures

Acronyms

Acronym	Description
AP	Average Precision
APTIV	Aptiv Services Deutschland Gmbh (EVENTS project partner)
AV	Automated Vehicle
CAV	Connected and Automated Vehicle
CP	Collective Perception
CPM	Collective Perception Messages
DNN	Deep Neural Network
Dx.y	Deliverable x.y
GNN	Global Nearest Neighbor
GNSS	Global Navigation Satellite System
EC	European Commission
ETSI	European Telecommunications Standards Institute
EXP	Experiment
FOV	Field of View
HIT	Hitachi (EVENTS project partner, includes both locations in France and UK)
ICCS	Institute of Communication and Computer Systems (EVENTS project partner)
INS	Inertial Navigation System
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
mAP	Mean Average Precision

Acronym	Description
MOT	Multi-object Tracking
MNIS	Multi-Target Normalized Innovation Squared
NEES	Normalized Estimation Error Squared
NIS	Normalized Innovation Squared
ODD	Operational Design Domain
PERCIV	Perciv.AI (EVENTS project partner)
REQs	Requirements
SA	Self-assessment
SAE	Society of Automotive Engineers
SOTA	State-of-the-art
SOTIF	Safety of the Intended Functionality
SPEC	Specification
TECN	Fundacion Tecnalia Research & Innovation (EVENTS project partner)
Tx.y	Task x.y
TUD	Technical University Delft (EVENTS project partner)
UC	Use Case
UULM	University of Ulm (EVENTS project partner)
V2X	Vehicle-to-everything
VRU	Vulnerable Road User
WMG	University of Warwick (EVENTS project partner)
WP	Work package

1. Introduction

1.1 Project aim

Driving is a challenging task. In our everyday life as drivers, we are facing unexpected situations we need to handle in a safe and efficient way. The same is valid for Connected and Automated Vehicles (CAVs), which also need to handle these situations, to a certain extent, depending on their automation level. The higher the automation level is, the higher the expectations for the system to cope with these situations are.

In the context of this project, these unexpected situations, where the normal operation of the CAV is close to be disrupted, - e.g. the Operational Design Domain (ODD) limit is reached due to traffic changes, harsh weather/light conditions, imperfect data, sensor/communication failures - are called “events”. EVENTS is also the acronym of this project.

Today, CAVs are facing several challenges (e.g. perception in complex urban environments, Vulnerable Road Users (VRUs) detection, perception in adverse weather and low visibility conditions) that should be overcome to be able to drive through these events in a safe and reliable way.

Within our scope, and in order to cover a wide area of scenarios, these kinds of events are clustered under three main use cases: a) Interaction with VRUs, b) Non-Standard and Unstructured Road Conditions and c) Low Visibility and Adverse Weather Conditions.

Our vision in EVENTS is to create a robust and self-resilient perception and decision-making system for Automated Vehicles (AVs) to manage different kinds of “events” on the horizon. These events result in reaching the AV limitations due to the dynamic changing road environment (VRUs, obstacles) and/or due to imperfect data (e.g., sensor and communication failures). The AV should have those events within its ODD and continue the operation safely. When the system cannot handle the situation, an improved minimum risk manoeuvre should be put in place.

1.2 Deliverable scope and content

Within EVENTS, WP3 addresses the development of the perception system, including localization and SA. The perception system consists of on-board perception (using camera, radar, and LiDAR sensors), which is supported by localization (using GNSS and INS), HD digital maps, and augmented by cooperative approaches (through V2X communication).

The objectives of WP3 are:

- Acquisition and adaptation of training data needed for machine learning-enabled perception systems to address the EVENTS use cases.
- Development of solutions for robust perception in complex urban traffic and urban area parks, which often feature a less structured road layout (e.g. unclear/non-existent road markings, narrow roads, and bridges). These settings might also be cluttered (e.g., infrastructure like traffic poles, lights and signs, or parked cars), and often involve close encounters with (possibly multiple) road users (e.g., VRUs), potentially approaching from various directions.
- Addressing the challenges of perception in poor visibility conditions due to lighting (e.g., night-time, blinding low-standing sun), adverse weather (e.g., rain, snow, fog), or other sensor impairments.
- Developing techniques for augmenting the on-board perception by using V2X information (e.g., Cooperative Awareness Messages (CAM), or Collective Perception Messages (CPMs) from other connected vehicles and/or from the road infrastructure.
- Development of methods for SA of perception systems that can detect deviations from the intended acceptable performance standards. These deviations may arise due to a variety of reasons, such as sensor impairments and noise, sensor de-calibration, faults in system components, or errors caused systems' misuses.

WP3 is structured in 5 sub-tasks (task leader is listed between brackets):

- Task 3.1 Training data acquisition and adaptation (ICCS)
- Task 3.2 Semantic scene analysis and precise localization (HIT)
- Task 3.3 Environment state estimation and motion prediction (TUD)
- Task 3.4 Augmented perception by V2X (UULM)
- Task 3.5 Perception system SA (WMG)

WP3 outputs will be used in WP4, as the decision-making and motion planning of WP4 strongly depends on the perception output. The validated perception system will be delivered to WP5 to be integrated in the overall EVENTS system.

Two Deliverables cover WP3 activities within the project: D3.1 and D3.2. The earlier submitted D3.1 [25] offered an *intermediate* snapshot of the work done in T3.1 – T3.4. The current D3.2 describes the *final* outcome of WP3, including T3.5 activities. This

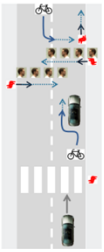
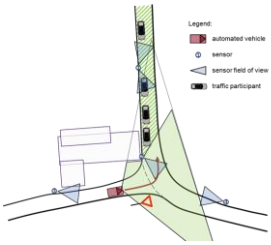
Deliverable mainly describes methodology and provides qualitative results. Rigorous quantitative evaluation is left for D6.2 ("Technical Evaluation Results").

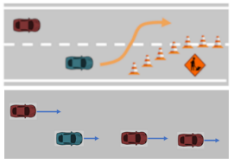
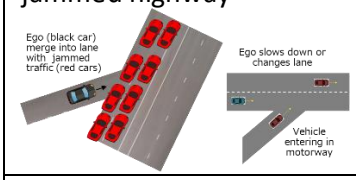
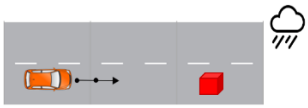
The remaining main sections of the document correspond to T3.1 and the EVENTS Experiments. The following sub-section recaps the latter.

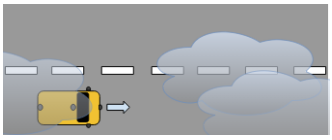
1.3 Experiments (EXPs)

Table 1 recaps the experiments that were selected for demonstration by the EVENTS consortium, as specified in Deliverable D2.1 **Error! Reference source not found.**. It provides the motivation for the various perception and localization approaches discussed in this Deliverable.

Table 1: Addressable experiments within EVENTS.

EXP1 - TUD Interaction with VRUs in complex urban environment 	EXP1 is about safe, comfortable, and time-efficient automated driving in complex urban environment while interacting with VRUs (e.g., pedestrians, cyclists). The environment perception, road user motion prediction, motion planning and vehicle control will be demonstrated in a single integrated system on-board TUD's own vehicle prototype. The experiment consists of the ego-vehicle driving on a two-lane road (i.e., one lane on each side) whereas several VRUs might (or might not) move into the vehicle's path (e.g., crossing, walk longitudinally, swerve), possibly from behind occlusions (e.g., parked vehicles). The question is whether to decelerate, accelerate, or steer away.
EXP2 – TECN, ICCS Re-establish platoon formation after split due to roundabout 	EXP2 incorporates perception augmentation via safe integration of collective perception (CP) info, predictive planning for the control of the platooning in an urban environment, management of the platooning behavior (T4.2) and design of a safe operational model for when an attached vehicle is in the platoon (T4.3). AV control takes advantage of augmented perception (inside and outside CAVs' FOV) offered by fusion of CAMs and CPMs (T3.4 and T3.5) shared by other road users and platoon members.
EXP3 - UULM Self-assessment and reliability of perception data with complementary V2X data in complex urban environments 	EXP3 is concerned with safe automated driving in a complex urban environment with occlusion, to demonstrate the integration of reliability assessment outputs of environment state estimation (onboard SA methods) and V2X data into an onboard perception system. The experiment will be conducted both in a virtual and a real environment. The former will be simulation-based, and it will be primarily concerned with developing a SA layer for the perception data (T3.5) along with complementary V2X data (T3.4). The latter will be realized in UULM's vehicle, with safety drivers/marshals to account for the prototypical status of the developed system, and in UULM's V2X infrastructure pilot site, where the automated ego-vehicle will face objects and (artificial) error/degradation in one of the sensors/V2X.

EXP4 – HIT-FR/UK, CRF, TECN Decision making for motion planning when faced with roadworks, unmarked lanes and narrow roads with assistance from perception self-assessment 	EXP4 is an end-to-end experiment starting with the precise vehicle localization, by defining a semantic representation of the environment (T3.2), and the motion prediction of dynamic objects in the scene (T3.3). The localization of the ego-vehicle will be further enhanced by using V2X information (CAM, CPM and Signal Phase and Timing (SPAT) messages, if available), thus increasing the reliability of its position in case of a failure or sensor blockage (T3.4).
EXP5 – HIT-FR/UK, CRF, TECN, WMG Decision making for motion planning when entering a jammed highway 	EXP5 is like EXP4 with two main differences. The first one is that there is a perception SA mechanism (T3.5). The second one is that the motion planning involves path and speed planning as well as control of the different highway entering experiments.
EXP6 - APTIV Small object detection at a far range in adverse weather conditions 	EXP6 concerns the sensing of small objects and semantic representation of these objects (relative position, height, object velocity, overdriveability and estimation of time to collision) within diverse weather conditions. In these situations, the object might not be clearly visible to the human eye and a critical decision on the vehicle behaviors needs to be taken. The vehicle should either avoid a potential frontal collision if the object is non-driveable by braking or avoid a potential rear collision with other vehicles driving behind if the object is over-driveable due to unnecessary braking.
EXP7 – ICCS, WMG Localization/perception self-assessment for advanced ACC and other vehicles' behavior prediction under adverse weather or adverse road conditions 	This experiment focuses on the development of an integrity monitoring mechanism for 3D LiDAR-based object detection and distance estimation to the leading vehicle in urban and highway environments under overcast and adverse weather conditions. The mechanism should reliably indicate the point in time when the relative localization of the ego-vehicle with respect to the leading vehicle must not be trusted and/or the object detection becomes unreliable. Another objective (not related with the SA objective) is to study the effects of adverse weather conditions on a perception module performing other vehicles' behavior prediction.
EXP8 - PERCIV Emergency evasion maneuver under adverse weather conditions	The low atmospheric visibility in adverse weather conditions like fog, snow, and rain reduces the maximum viewing distance of LiDAR sensors. This in turn decreases the object detection and localization performance and cause safety hazards. Weather conditions have effect on sensing and

including perception self-assessment 	therefore on perception and localization of automated driving system. This use case provides the possibility to evaluate the on-board visibility-based localization performance estimate. Safe vehicle control is necessary in case the weather conditions worsen and fail-safe behavior in case of exiting the ODD completely due to extreme weather.

2. Training data acquisition and adaptation

This section reports on the progress made by the EVENTS partners on activities pertaining to Task 3.1. The report is organized along the directions set in D3.1 [25], specifically:

i. Data generation and augmentation (ICCS, HIT).
ii. Exploration and utilization of (experiment-specific) public datasets (TECN for EXP2).
iii. Acquisition of a new road debris dataset within EVENTS (APTIV).
iv. Self-Supervised Learning (TUD).

Each of the following sections focuses on progress on the above topics, as conducted by the respective involved partners. With this report, the work of Task 3.1 has been successfully concluded. This work has partially fed the work in T5.1 (scenario editing for simulation and data logging, see D5.1 [157]) and T6.2 (data preparation for evaluation) respectively.

2.1 Data Generation and Augmentation

2.1.1 Adverse weather image translation

Generating synthetic driving-related images with adverse weather conditions at scale by means of deep generative models is challenging, as already described in D3.1 [25]. Apart from the training instability characterizing GANs in general, retraining image-to-image translation models for specific translations (rain, snow, fog) has intense computational requirements, even for producing images at low resolutions (256x256, 256x512) [1], [2], [3], [4]. ICCS has further experimented with off-the-shelf models, namely pix2pix and Stable Diffusion. The translated images were frequently of acceptable visual quality (Figure 1), but still required manual inspection for reassuring usability (Figure 2). Furthermore, the annotations of the original image were not guaranteed to hold for the translated one, a problem most often observed in the Stable Diffusion model (Figure 3, Figure 4). Overall, the explored approaches based on GANs do not appear to alleviate significantly the manual effort required to produce large scale quality datasets.

Figure 1: Original image (left), synthesized via pix2pix (right).

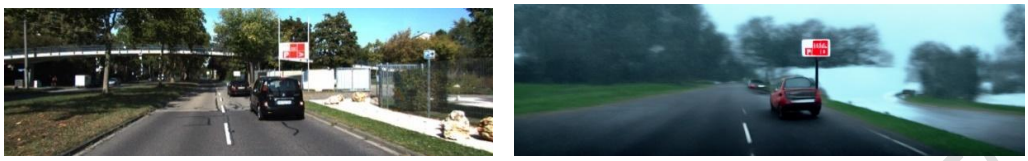


Figure 2: Original image (left), synthesized via pix2pix (right).

Figure 3: Original image (left), synthesized via pix2pix (right).

Figure 4: Original image (left), synthesized via Stable Diffusion (right).

Therefore, a further study of the effects of artificial data augmentation was conducted along the following two axes:

a) A study on the effects of augmenting the CityScapes dataset (by reproducing [4]) on the small, medium, and large versions of the state-of-the-art YOLOv8 object detector. The test set was a custom dataset of 10.000 images selected from the Canadian Adverse Driving Conditions Dataset. The detectors were retrained on the following data: (i) the original CityScapes (ii) both the original CityScapes and all of its weather-translated versions (iii) same as in (ii) without the original dataset, and evaluated by the mAP 50-95 metric. Results are depicted in the table below. Overall, augmenting the dataset was beneficial only for the larger models, but even for them not significantly.

Model/ Training set	CityScapes	CityScapes + Translations	Translations
Yolov8 Small	0.40	0.40	0.39
Yolov8 Medium	0.44	0.46	0.45

Yolov8 Large	0.46	0.49	0.49

b) A study on the effect of augmenting a part of nuscenes [7] by random global and local blurrings and perspective transformations to emulate corrupted visibility on the same object detector as in (a). No significant improvements were observed for detectors of all sizes (a max of ~0.05 in every experiment).

Based on the above observations and results, ICCS shifted the main effort towards simulated data generation.

2.1.2 Data generation via simulation

Two distinct software frameworks (besides pytrees) were explored for scenario generation and variation in CARLA by ICCS, namely (i) Roadrunner[1] and (ii) Scenic[2]. Additional effort was put for the integration of RGB and Lidar sensors on the involved actors.

Pytrees is a Python library consisting the standard set of tools for scenario generation and editing in the CARLA simulator. Although it is readily compatible with the simulator, it requires coding skills and both the creation and variation of scenario are quite time consuming. Furthermore, the scenarios produced by pytrees can be used only by CARLA, strictly excluding general applicability.

An alternative and most commonly used approach is based on the ASAM OpenScenario XML standard for scenario specification [8]. OpenScenario formalizes the scenario description language via XML semantics in a universal format consumable by driving simulators. It is fully interoperable with the ASAM OpenDrive standard for specifying constant map elements (Figure 5) [9]. The Roadrunner software supports both standards and facilitates scenario generation and variation via a user-friendly graphical user interface and its gRPC API. The entire stack was tested on specifying and varying two distinct scenarios, namely a motorway cut-in scenario and platooning in a roundabout. In both scenarios, initial specification and slight variations were easily implemented. In both cases, some additional scripting transforming the produced files to a CARLA Scenario Runner consumable format was needed, an effort expected to be minimized in future versions of the software. Imposing slight variations on both scenarios was perfectly feasible by simple scripting. Therefore, exploiting Roadrunner and the CARLA ground truth appears as a valuable toolchain for multiple view simulated data generation (Figure 6). However, OpenScenario limits the specification of vehicle trajectories in defining simple path following actions (Figure 7) without the option of specifying additional control flow on top of them. This limitation appeared

[1] https://www.mathworks.com/products/roadrunner.html (free for academic use)
[2] https://docs.scenic-lang.org/en/3.x/publications.html

quite restrictive while varying platooning scenario, pointing to a rather limited usability for varying more complex scenarios in general.

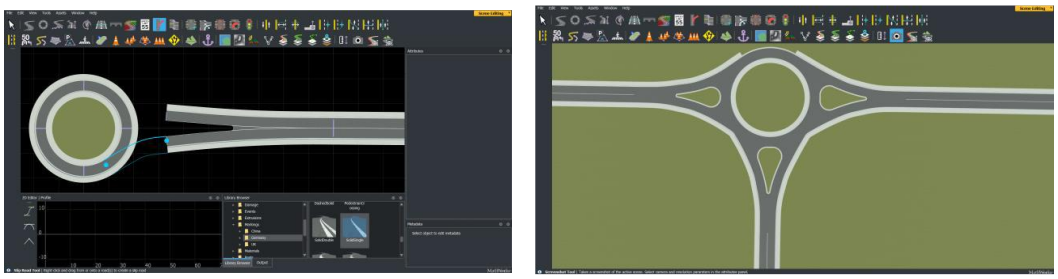


Figure 5: Map Construction (left) and final form (right) - snapshot from RoadRunner map creation

Scenic is domain-specific scripting programming language for modeling scenarios involving agent (e.g. robots or vehicles) movement and interactions [10]. It is intrinsically integrated with a variety of simulators including CARLA, thus the resulting coded scenario can be readily executed in any compatible simulator without the need of additional integration effort (Figure 8). Scenario parameters are specified in ranges; starting a simulation will randomly pick parameters within the specified ranges, thus a large number of variations can be simulated in a straightforward manner. Apart from that, Scenic allows the definition of actor behavior via built-in or custom specified conditional control flows, a feature the OpenScenario stack is still lacking. Therefore, Scenic enables the specification and simulation of a large variety of actor interactions, including behaviors reacting dynamically to the current environment.

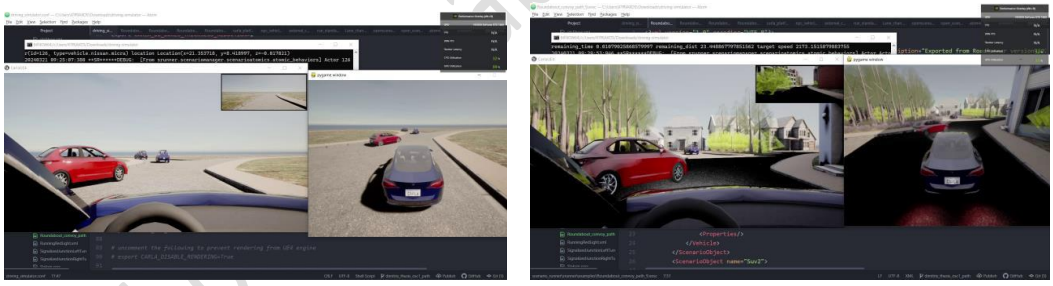


Figure 6: Multiple view simulation of a custom scenario in different maps (snapshots from CARLA build)

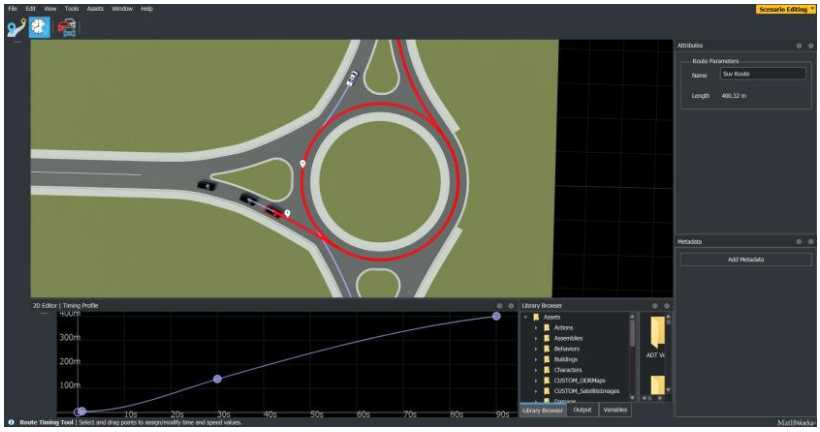


Figure 7: Definition of vehicle trajectory via space-time waypoints (snapshot from RoadRunner scenario creation).

Apart from exploring the aforementioned features, modifications of the Scenic source code were carried out in order to:

(i) spawn multiple sensors on the participating agents,

(ii) log the agents' ground truth data including locations, velocities and bounding boxes and sensor recordings during simulation runtime,

(iii) project 3D bounding boxes to any camera frame (via the appropriate coordinate transforms) and display during runtime (for an example visualization, see Figure 9)

(iv)

Figure 8: Pedestrian crossing behind obstacle, oncoming car from opposite lane (snapshot from Scenic-CARLA co-simulation environment).

Figure 9: Oncoming car ground truth bounding box display during runtime (snapshot from Scenic-CARLA co-simulation environment)

2.1.3 Annotated traffic sign dataset generation via patch augmentation

A challenge that arises in traffic sign recognition is the ability to recognize non-standard or country-specific signs. Examples of non-standard traffic signs include signs with specific semantic meaning (e.g., road works) and country-specific signs in Japan (e.g. the road work sign in Japan).

The public datasets related to traffic signs explored by HIT (refer to D3.1 [25]), can be grouped as follows: 1) datasets that are generated by the public (and labelled by human annotators), e.g., the Mapillary dataset [11], and 2) an AV specific dataset that was generated primarily for European roads, i.e., Zenseact dataset [12]. While both datasets are valuable for training a sign detector, the aforementioned issues remain, i.e., the lack of generalization to non-standard and country-specific signs. To this end, we propose to patch augment a template sign onto an existing public dataset to generate a labelled dataset that can be used to train an object detector. Prior works in this area have shown such approaches are effective in boosting performance of baselines models by patch augmenting classes onto images [13], [14].

Our proposed method is relatively simple and patches an object (to be detected) onto an image without considering if the object placement in the scene is realistic. To capture the camera viewpoint perspective distortion and scaling effects, we propose to use projective geometry and image scaling respectively. Finally, to capture variations in intensity, we use conventional intensity augmentations such as affine transformations of pixels.

To summarize, our proposed method consists of the following steps:

- Identify a public dataset with significant scene variation. Collect a database of traffic signs, which can be sourced from the internet.
- Randomly select a traffic sign for patching onto an image from the public dataset.
- Randomly resize the selected traffic sign and apply data augmentation techniques (pixel geometry transformations and pixel intensity transformations).
- Identify a random patch location within a selected image from the public dataset and patch onto it the transformed traffic sign.

An example patch augmented image using the proposed approach is shown in Figure 10.

Figure 10: Patch augmentation on public dataset not considering both traffic sign configuration and camera viewpoint.

2.2 Exploration and use of public datasets for cooperative motion prediction

TECN develops a cooperative motion prediction module to be used within the framework of EXP2. To this end, a variety of publicly available datasets has been explored, including datasets most commonly used in the literature for this purpose like nuScenes [15], Waymo [16], and Argoverse 1 [17]. Despite their popularity, these datasets do not consider information from associated perceptual providers. To address this challenge, several additional datasets have been explored that capture perceptual systems from different perspectives. These datasets are depicted in the table below.

Table 2: Comparison between Cooperative Perception-related Datasets.

Dataset	Real/Sim	V2X	Size (km)	LiDAR pcds	Map	3D Boxes	Classes	Locations
OPV2V [18]	Sim	V2V	-	11k	Yes	230k	1	CARLA
V2X-Sim [19]	Sim	V2V&I	-	10k	Yes	26.6k	1	CARLA
V2XSet [20]	Sim	V2V&I	-	11k	Yes	230k	1	CARLA
A9 Intersection [21]	Real	V2I	-	4.8k	No	57.4k	10	Hanover, Ger
DAIR-V2X [22]	Real	V2I	20	39k	No	464k	10	Beijing, CN
V2X-Seq [23]	Real	V2V&I	-	210k (seq)	No	20k (2D)	8	Beijing, CN
V2V4Real [24]	Real	V2V	410	20k	Yes*	240k	5	Ohio, USA

Datasets [18][19][20] are simulated, and hence are rather simplistic in terms of vehicle physics and sensor models. The A9 dataset [21] collected 4.8k frames from two cameras and two LiDARs placed on infrastructure. However, it does not provide information from connected vehicles. DAIR-V2X [22] was the first large-scale vehicle-infrastructure cooperative autonomous driving dataset. It collects data from the infrastructure (10k frames) and a vehicle (22k frames). Incidentally, the data must be approved to be downloaded outside of China. Recently, the V2X-Seq dataset [23] was released. It is the first large-scale sequential V2X dataset. The temporal information

makes it perfect for motion prediction. However, like DAIR-V2X, its access outside China is restricted.

For these reasons, the dataset primarily used for the evaluation of TECN's cooperative motion prediction module was V2V4Real [24]. V2V4Real consists of data from two vehicles recorded in real environment. It does not provide sequence information, but the dataset is prepared for tracking purposes, so it was further (pre) processed to extract sequence information.

2.3 Acquisition of a new road debris dataset

A new road debris dataset was collected in order to obtain the required training and test data for the perception algorithm used in EXP6. A literature review on debris-related accidents was conducted in [25] providing valuable insights on the selection of objects to be included in this new dataset.

Data used for selecting and training a method for object classification was collected in a controlled and repeatable manner. The data was obtained from a front-facing radar, mounted on the host vehicle. Objects are placed ahead of the host vehicle at distances depending on the test speed, approximately 500 m for a speed of 130 kph and around 350 m for lower speeds. During data collection, the host vehicle approaches the object in a straight line accelerating to reach the target speed. Once the desired speed was reached, the vehicle began decelerating, coming to a complete stop, directly in front of the debris.

The data was collected using different speeds: 50, 90 and 130 kph with the data recording starting at 300 m (refer to Figure 11).

The object orientation was also varied during data collection for those objects that had a major and minor axis. Depending on their shape, objects were placed facing the host vehicle (0° orientation) or at 45° and 90° rotation angle (refer to Figure 12).

Objects were placed either directly in the middle of a path of the host vehicle or with a lateral offset of 1.5 m (refer to Figure 13).

Data was collected for 47 different objects. Due to time constraints on the test track, rotated and lateral offsets were only varied when the host vehicle speed was 50 kph.

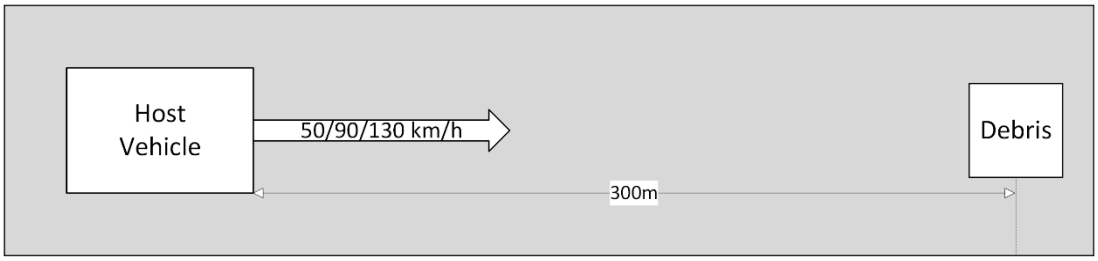


Figure 11: Measurement: object on the path. Varied velocity.

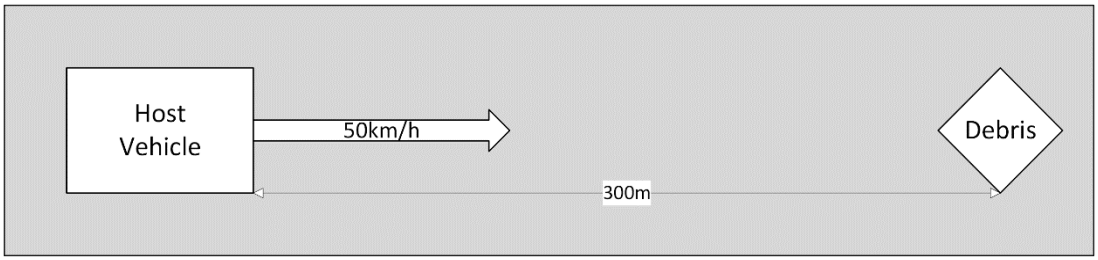


Figure 12: Measurement: Object on the path. Rotated.

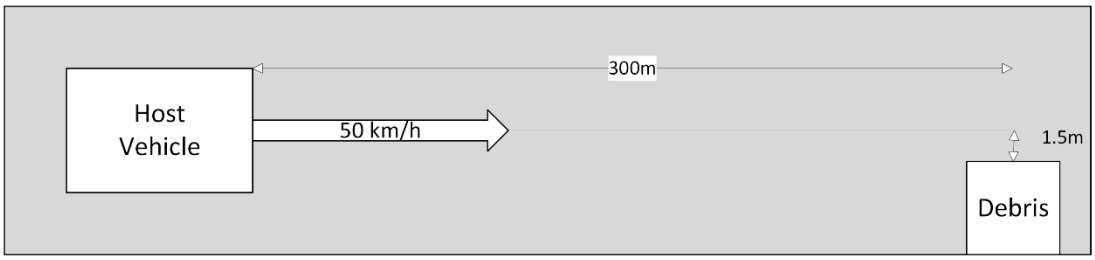


Figure 13: Measurement: Object with lateral offset to the path.

2.4 Self-Supervised Learning

In this section, the contributor from the EVENTS' partners is TU Delft.

Object detection is one of the core tasks of computer vision, and it is integrated into the pipeline of many applications such as autonomous driving [26], person re-identification [27], and robotic manipulation [28]. During the past decade, the computer vision community has made tremendous progress detecting objects, especially learning-based methods. These supervised methods rely on manual annotations, i.e., each object instance is indicated by a bounding box and a class label. However, a massive amount of labeled training data is usually required for training those models, while labeling is expensive and laborious. This raises the question of how object detection models can be trained without direct supervision from manual labels.

Unsupervised object detection is a relatively unexplored research field compared to its supervised counterpart. For camera images, recent work [29], [30] shows that the emergent behavior of models trained with self-supervised representation learning can

be used for object discovery, i.e. object localization without determining a class label. The behavior implies that the learned features of those models contain information about the semantic segmentation of an image, and thus, they can be used to distinguish foreground from background. Consequently, the extracted coarse object masks are used to train 2D object detectors [31], [32]. Although these methods perform well for images depicting a few instances with a clear background, they fail to achieve high performance for images with many instances, such as autonomous driving scenes [33]. In these scenes, instances are close to each other and, as a result, are not directly separable using off-the-shelf features.

On the other hand, spatial clustering is the main force that drives 3D object discovery [33], [34]. In contrast to images, separating objects spatially is relatively easy in 3D space, but differentiating between clusters based on shape is challenging because of the density of the data (e.g. sparse LiDAR point clouds). Hence, temporal information is often exploited to identify dynamic points that most likely belong to mobile objects such as cars and pedestrians. In this context, we define mobile objects as objects that have the potential to move. Consequently, objects such as buildings and trees are considered non-mobile classes. The discovery of static foreground instances (e.g. parked cars and standing pedestrians) is usually achieved by performing self-training. Self-training is based on the assumption that a detector trained on dynamic objects has difficulty discriminating between the static and dynamic versions of the same object type. As a result, when such a detector is used for inference, it will also detect many static instances. The predicted objects are then used for retraining the detector, i.e. self-training, which is repeated multiple times until performance converges.

We argue that multi-modal data should be used jointly for unsupervised 3D object discovery as each modality has its own strengths, e.g. cameras capture rich semantic information and LiDAR provides accurate spatial information. Existing work [33] does use multi-modal data for unsupervised object discovery but not jointly. The training procedure consists of two parts: (1) training with LiDAR-based pseudo-bounding belonging to dynamic instances and (2) multi-modal self-training to learn to detect static and dynamic objects. However, Wang et al. [33] ignore the fact that both modalities can be used at the same time for creating pseudo-bounding boxes.

2.4.1 Method

We propose our method, UNION [35], that exploits the strengths of camera and LiDAR jointly, see Figure 14. We extract object proposals by spatially clustering the non-ground points from LiDAR and leverage camera to encode the visual appearance of each object proposal into an appearance embedding. Subsequently, we exploit the appearance similarity between static and dynamic foreground objects for discriminating between static foreground and background instances. Finally, the identified objects and their appearance embeddings are used to generate pseudo-

bounding boxes and pseudo-class labels, which can be used to train existing 3D object detectors in an unsupervised manner using their original training protocol.

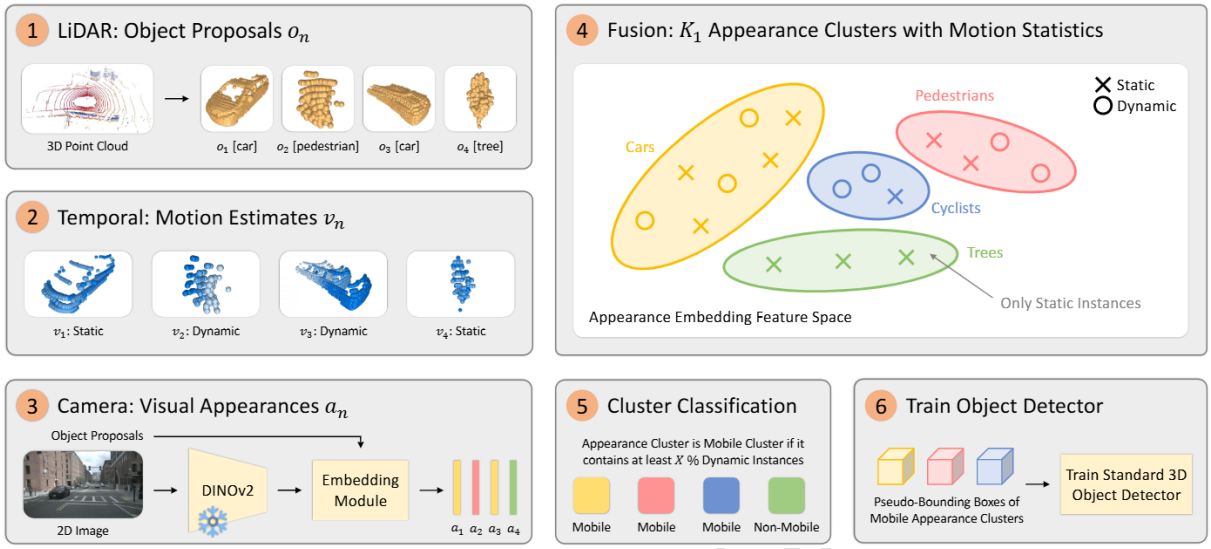


Figure 14: UNION discovers mobile objects in an unsupervised manner by exploiting LiDAR, camera, and temporal information jointly. The key observation is that mobile objects can be distinguished from background objects by grouping object proposals with similar visual appearance and selecting appearance clusters that contain at least X % dynamic instances.

2.4.2 Dataset

We evaluate our method on the challenging nuScenes [36] dataset. This is a large-scale autonomous driving dataset for 3D perception captured in diverse weather and lighting conditions across Boston and Singapore. It consists of 700, 150, and 150 scenes for training, validation, and testing, respectively. A scene is a sequence of 20 seconds, and is annotated with 2 Hz. Each frame contains one LiDAR point cloud and six multi-view camera images.

2.4.3 Intermediate Representations UNION

The first step for generating object proposals is to extract the non-ground points from all LiDAR point clouds as the non-ground points may belong to mobile objects. The non-ground points are spatially clustered to get object proposals, i.e., 3D segments. Step 1 in Figure 14 illustrates the generation of these class-agnostic 3D object proposals, and Figure 15 and Figure 16 show the ground segmentation and spatial clustering for a sample of the nuScenes [36] dataset, respectively.

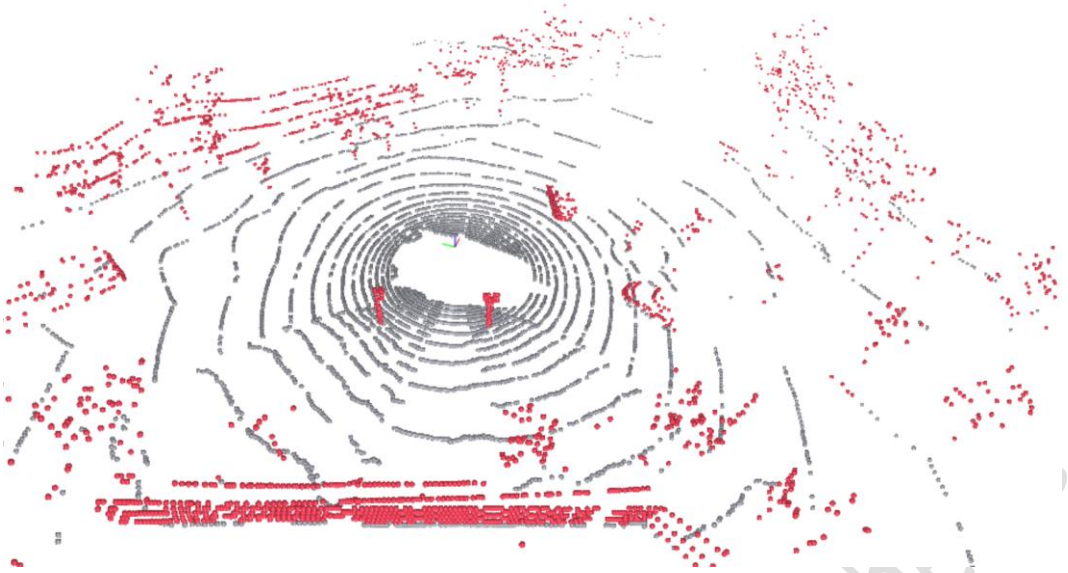


Figure 15: A LiDAR point cloud segmented into ground (gray) and non-ground points (red).

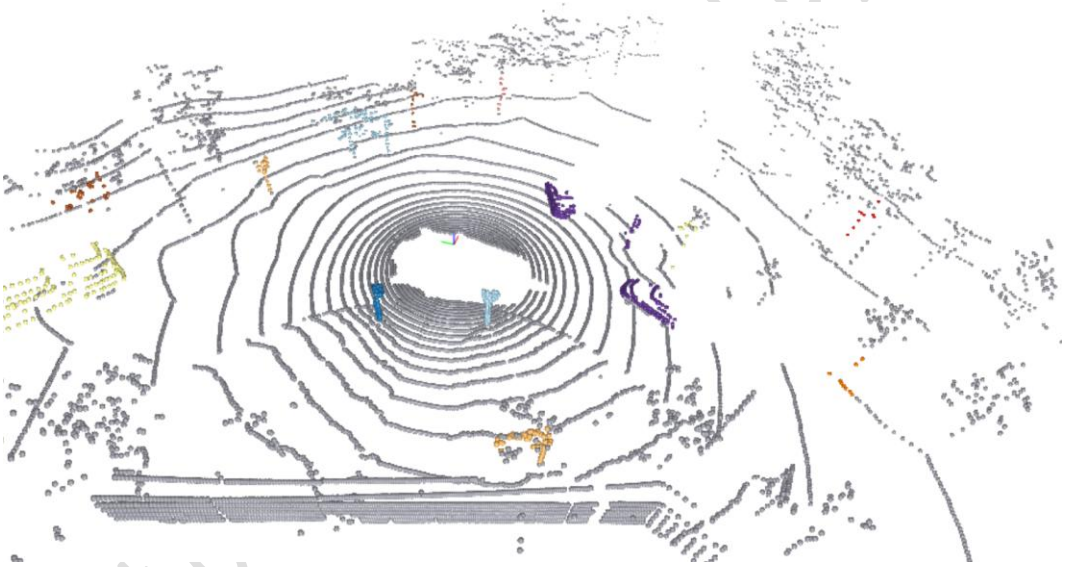


Figure 16: The spatial clusters of the non-ground points from Figure 2. The ground points are indicated by gray, while the spatial clusters (object proposals) have non-gray colors.

We estimate the motion status of the object proposals to determine whether each proposal is static or dynamic. The object proposals contain temporal information as the non-ground points from multiple time steps have been aggregated before the spatial clustering. In other words, the motion can be observed when the 3D points of an object proposal are split into different sets based on their time step, i.e., undoing the aggregation. This is shown by step 2 in Figure 14, and Figure 17 shows the dynamic object proposals for a sample of the nuScenes [36] dataset.

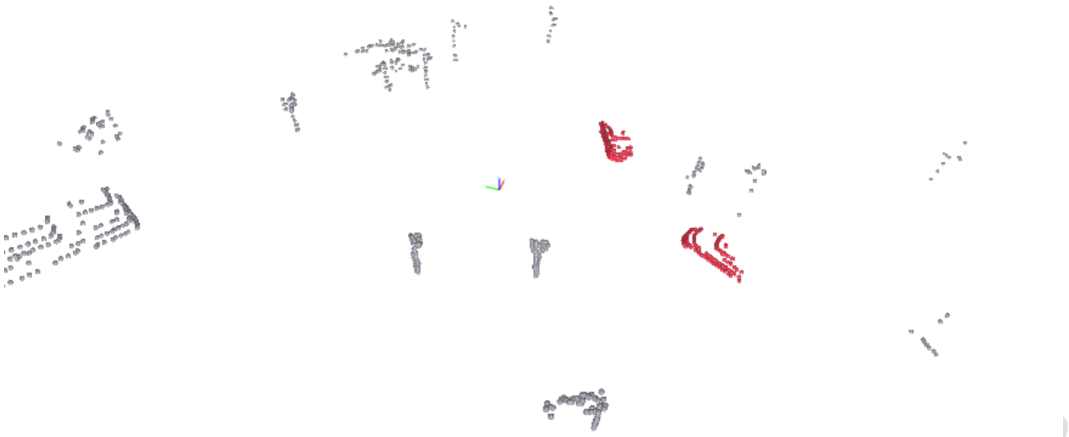


Figure 17: Dynamic object proposals (red) and static object proposals (gray).

2.4.4 Generated Pseudo-Bounding Boxes

Figure 18 provides qualitative results of the generated pseudo-bounding boxes for an example scene of the nuScenes [36] dataset. The ground truth boxes are also shown. It can be seen that the scene flow can identify multiple dynamic objects, and the appearance clustering can discover static mobile objects, including vehicles and pedestrians, using those dynamic instances.

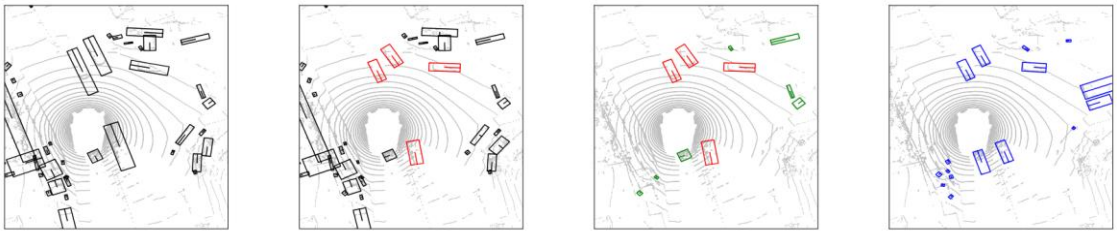


Figure 18: Qualitative results for the UNION pipeline compared to the ground truth annotations. (a) HDBSCAN [37] (step 1 in Figure 14): object proposals (spatial clusters) in black. (b) Scene flow (step 2 in Figure 14): static and dynamic object proposals in black and red, respectively. (c) UNION: static and dynamic mobile objects in green and red, respectively. (d) Ground truth: mobile objects in blue.

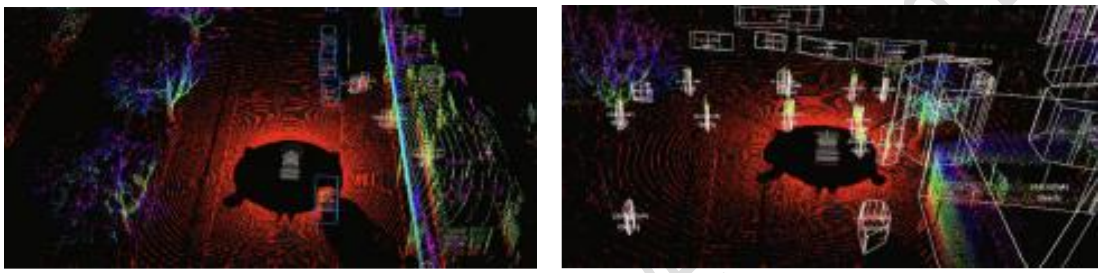
3. WP3 Modules in EVENTS Experiments

3.1 EXP1 (TUD): Interaction with VRUs in complex urban environment

EXP1 is about safe, comfortable and time-efficient automated driving in complex urban environment while interacting with VRUs (e.g., pedestrians, cyclists). We use a modular LiDAR perception pipeline to detect and track objects, as well as predict their future motion.

3.1.1 Overview

We fuse the point clouds from both LiDARs to generate a high-resolution ego-motion compensated point cloud that covers the 360 surroundings of the vehicle. To detect objects from a set of predefined classes, as well as generic objects, we use two detection components. A deep learning-based detector detects objects from a set of predefined classes, e.g., car, bike, and pedestrian. We also use a traditional LiDAR clustering pipeline to segment generic objects that are not part of the above set of classes. An object merger combines the detections of both methods and discards near-identical duplicate detections. A multi-object tracker tracks detected objects across time, and a motion prediction component predicts the future path of each tracked object.



(a) (b)

Figure 19: A visualization of the outputs of the object detector and LiDAR clustering. (a) The object detector detects a predefined set of object classes (e.g. car and pedestrian), cf blue bounding boxes. (b) The LiDAR clustering segments any object protruding from the ground plane, including generic objects, cf. white bounding boxes.

3.1.2 Object Detection

We use the deep learning-based CenterPoint [38] detector for LiDAR-based 3D object detection, since it is accurate and can run in real-time. Autoware [45] provides the model parameters for CenterPoint trained on the nuScenes [36] dataset for autonomous driving. However, the LiDAR used in nuScenes is significantly sparser than our LiDAR, which results in a drop of the detection rate. Therefore, we retrain CenterPoint on the recently published Zenseact [39] dataset to improve the detection performance. This dataset was captured using a LiDAR with a similar density as our LiDAR, and as a result, the smaller domain shift between the source dataset and our demo environment results in a higher detection performance than achieved with CenterPoint trained on nuScenes. Figure 19a shows the detected objects for a scene.

Within the quantitative evaluation in WP6, we aim train the CenterPoint detector on a larger dataset obtained by our self-supervised framework described in Section 3.2.4.

LiDAR clustering. Given a LiDAR point cloud, we remove the ground plane [40] and group the remaining points into several class-agnostic clusters [41]. Then, we fit an L-

shaped bounding box for each cluster using [42]. We tune the clustering hyperparameters on a small set of self-collected data near the campus by examining the result qualitatively. Figure 19b shows the clustering results for a scene.

Object Merging. The object detector and LiDAR clustering step can output duplicate detections for the same physical object. Merging these detections can be regarded as a minimum-cost flow problem (MCFP). We solve the MCFP using the successive shortest path (SSP) algorithm [43]. Matched objects are assigned labels derived from the CenterPoint detection pipeline. Detections that do not have a corresponding match are labeled as "unknown objects".

3.1.3 Multi-Object Tracking.

We apply a multi-object tracker on the merged objects to smooth tracks, infer the track identities, and estimate the velocity of each object. The multi-object tracker consists of a data association module and an Extended Kalman Filter (EKF) tracker. The data association module performs maximum score matching to associate the objects from neighboring frames. Our implementation uses muSSP [44] as the solver to achieve real-time performance.

We build a separate EKF tracker for each learned object class, as well as the unknown object class.

3.1.4 Motion Prediction

We currently use the EKF proposed in the previous step and propagate the uncertainty 4 seconds into the future. These unimodal distributions represent the future locations of all actors in the scene. The predicted distributions are fed into the motion planner (cf. WP4).

Within WP5, we aim to integrate the more sophisticated map-based PGP motion-prediction method discussed in [156], into the vehicle, which allows multi-modality in the prediction.

3.2 EXP2 (ICCS, TECN): Re-establish platoon formation after split due to roundabout

3.2.1 Introduction

EXP2 is based on the following rationale; (i) Platoon re-establishment is a long standing focus in the CCAM research community. (ii) Methods for trajectory prediction of non-ego vehicles have undergone important advances recently. (iii) The potential of V2X augmented (collective) perception remains quite unexplored across a variety of traffic scenes and respective occlusion scenarios. Thus, EXP2 aims to explore the combined

potential of trajectory prediction and V2X-augmented collective perception in the re-establishment of a vehicle platoon within a roundabout setting.

3.2.2 Cooperative Motion Prediction

Motion prediction

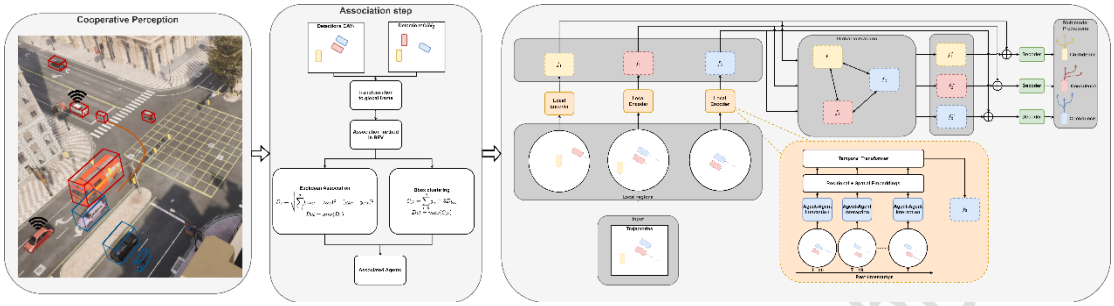


Figure 20: Cooperative Framework for Motion Prediction.

TECN develops a cooperative motion prediction framework [57] that considers all perceived vehicles to compute the future trajectories of the surrounding agents. Its proposal is based on Hierarchical Vector Transformer (HiVT) [58], without a specified map for enhanced generality. The model was trained on the Argoverse 1 dataset [48]. The model's input is a collection of vectorized entities representing the traffic scene. Agent interactions are encoded by a local encoder, which enables a rotation-invariant representation for each agent for scalability and better learning. A subsequent global module encodes the long-range dependencies between the agent-centric local representations. Finally, a multimodal future decoder predicts the trajectories of all agents in a single pass. The output is represented on a CAV local coordinate frame. Planned future work includes varying this frame by frame, to measure the impact of changing viewpoints on the predictions.

Table 3: Performance of the motion prediction model in Argoverse 1.

Model	Map	minADE (m) ↓	minFDE (m) ↓	MR (%) ↓
HiVT-64 [13]	✓	0.69	1.04	0.1
HiVT-128 [13]	✓	**0.66**	**0.96**	**0.09**
Crat-Pred [14]	✗	0.85	1.44	0.17
HiVT-64 (ours)	✗	0.76	1.24	0.14

Association methods

Multiple overlapping detections of the same object by different CAVs should be combined to improve accuracy and reliability of the motion prediction framework. To this end, we have tested two available methods. The first method resolves overlapping detections by merging them to the detection obtained by the closest CAV, in terms of Euclidean distance. The second method counts the number of lidar points within each detected bounding box and selects the bounding box containing the maximum

number of LiDAR points, which implicitly considers occlusions and is therefore more accurate. A future version of the motion prediction network will also consider the detector confidence.

Experimental setup and results

The model was evaluated on the complete V2V4Real [55] dataset on a total of 20,000 frames by means of Brier Scores for minADE and minFDE [48][58]. Initially, the model was evaluated without V2V enhanced perception. The next evaluation assumed V2V without or with Euclidean/bbox clustering associations. Subsequently, following [60], we changed the viewpoint for the motion prediction module. The performance evaluation results are depicted in Table 4.

Table 4: Comparison of methods on the V2V4Real dataset normalised by the number of actors in the scene. We show the CAVs, the association method, the viewpoint and the performance metrics. The"-" denotes that there is no association method used.

CAVs	Fusion	ViewPoint	Num actors	brier-minADE (m)			brier-minFDE (m)		
				Absolute	Relative	Improvement	Absolute	Relative	Improvement
Tesla	-	Tesla	7.74	**1.80**	0.23	-	**2.88**	0.37	-
Astuff	-	Astuff	8.45	1.88	0.22	-	2.96	0.35	-
Tesla & Astuff	-	Tesla	14.58	2.00	**0.14**	-	3.17	**0.22**	-
Tesla & Astuff	Euclidean	Tesla	10.13	1.92	0.19	18%	3.02	0.30	20%
Tesla & Astuff	Bbox clustering	Tesla	10.19	1.92	0.19	19%	3.03	0.30	20%
Tesla & Astuff	-	Astuff	14.58	2.00	**0.14**	-	3.18	**0.22**	-
Tesla & Astuff	Euclidean	Astuff	10.13	1.92	0.19	**15%**	3.04	0.30	**14%**
Tesla & Astuff	Bbox clustering	Astuff	10.19	1.93	0.19	**15%**	3.05	0.30	**15%**

Since an increased number of perceived actors is expected to raise the error, the "Relative" column depicts the ratio Absolute/number of actors. Apparently, the V2V enhanced perception improves performance. No significant differences were observed by altering the association methods and changing the model's field of view. Indicative qualitative results are depicted in Figure 21, Figure 22, Figure 23, Figure 24, Figure 25, and Figure 26.

We represent: the Tesla point cloud, the Astuff point cloud, the agents, the past observations, the ground-truth and our multi-modal prediction (with the highest confidence). We show, from left to right, single-vehicle, Euclidean and bbox clustering.

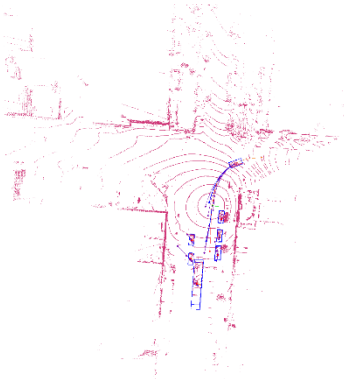


Figure 21: Single-vehicle Tesla.

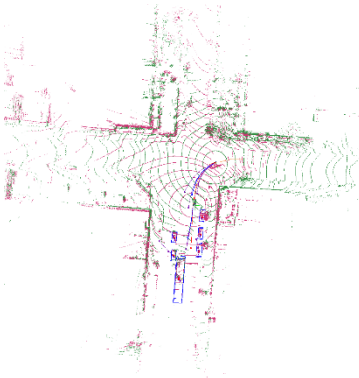


Figure 22: Euclidean association with the tesla as viewpoint.

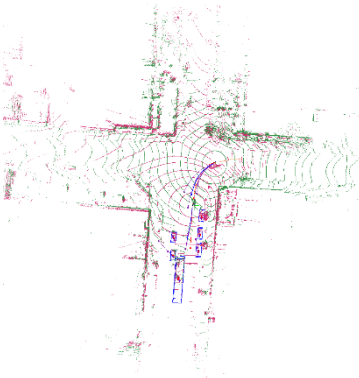


Figure 23: Bbox clust. association with the Tesla as viewpoint.

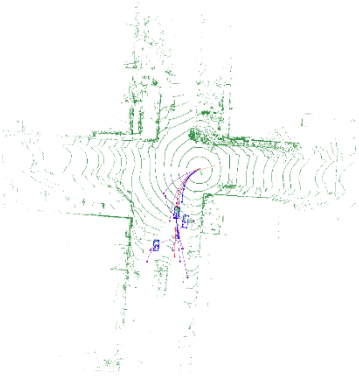


Figure 24: Single-vehicle Astuff.

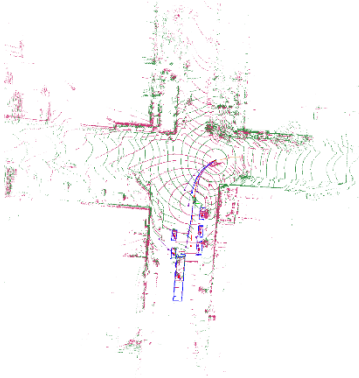


Figure 25: Euclidean association with the Astuff as viewpoint.

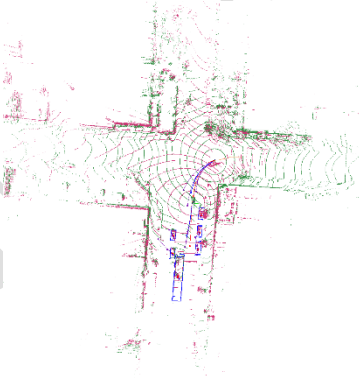


Figure 26: Bbox clust. association with the Astuff as viewpoint.

3.2.3 Augmented Perception via V2X-CAM-CPM

The formalization of CAM and CPM messages was based on the specifications provided in the respective ETSI documents [61], [62], [63], [64]. JSON file formats were specified accordingly, where each data field is defined in terms of data type (e.g., number, string etc.) and physical interpretation and units were applicable. The CAM JSON file includes all relevant data on the state of the ego-vehicle including position, heading, yaw rate, speed, acceleration and vehicle dimensions. The CPM provides details for each perceived object including object type, vertical and horizontal distance and speed coordinates (w.r.t. to the ego-vehicle), yaw angle and planar object dimensions.

In the EXP2 setup, three CAVs perform platooning manoeuvres alongside simulated dummies within the CARLA simulator. The 3D detections from these scenarios are parsed into CAMs and CPMs, formatted using the ROS standard message type **std_msgs/String.** The information included in the CAM and CPM messages disseminated by each vehicle is used by ICCS to construct a collective estimation of

the bird's eye view of the scene in terms of a probabilistic occupancy grid as already described in D3.1 [25].

Algorithm overview

Collective perception of CAVs is a relatively recent research area, for which a variety of approaches have been proposed, ranging from methods motivated by mobile robotics to more sophisticated deep-learning based techniques [49], [50], [51], [65], [66], [67], [69]. The adopted approach focuses on: (1) Use of CAM/CPM; (2) End-to-end explainability; (3) Algorithm parameters that correspond to observable physical quantities that can be measured and/or deduced; (4) Using evaluation measurements/metrics characterizing individual perception reliability of the actors involved and fusing them in a statistically transparent and probabilistically sound way; (5) Providing inherent and intuitive ways of (a) checking the consistency of received individual perception data, and (b) deriving metrics of output reliability. The overarching architecture is depicted in Figure 27.

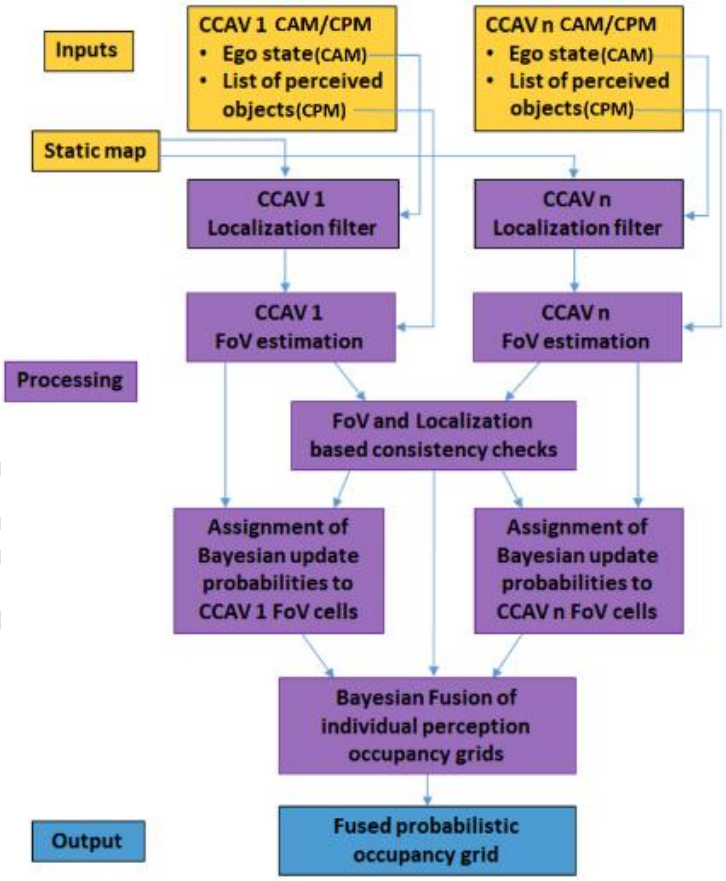


Figure 27: Algorithm Overview.

CAV Localization Filters

We consider a non-linear, Ackermann-type model for spatiotemporal vehicle dynamics [68]:

$$\dot{x} = vcos\theta - vtan\varphi sin\theta/2$$
$$\dot{y} = vsin\theta - vtan\varphi cos\theta/2$$
$$\dot{\theta} = (v/l)tan\varphi$$

where x, y are the cartesian coordinates of the center of the CAV axis, θ is the heading angle, φ is the Ackermann steering angle, l denotes the distance between front and back wheel axes and v denotes speed. The corresponding measurement model considers additive Gaussian noise for each one of the motion variables x, y, θ based on the respective device measurement error. Particle-based localization filters [69], [70] have been designed, achieving a 0.1 mean absolute estimation error for $\sigma_x, \sigma_y = 0.33\ m$ (corresponding to a $3\sigma \approx 1\ m$ standard GPS error), $\sigma_\theta = 5^o$ and 100 particles.

CAV Field of View calculation and Bayesian fusion

The Field of View (FoV) of each CAV is calculated according to its disseminated CPM/CAM. The FoV calculation method is based on a custom ray-casting approach that has been redesigned to consider both spatial dimensions and orientation of the involved vehicles (Figure 28). Additionally, significant effort was spent on GPU implementation. As described in D3.1 [25], the calculated FoVs combined with forward sensor model probabilities are subsequently used for Bayesian fusion and derivation of the resulting probabilistic occupancy grid.

Online reliability assessment

The described approach lends itself to straightforward and intuitive derivations of quantitative indicators for assessing the reliability of the output. Specifically:

Starting from the CAVs localization step, a set of reliability indicators for the step's output can be derived from the estimated covariances of the posterior state estimations of the filter recursions. These covariances characterize the uncertainty ellipse around each estimated CAV position. In case of Kalman filters, these are the resulting covariance matrices; in [71] the authors use these matrices for conducting pertinent statistical tests. In case of a particle filter, the resulting covariances can be directly calculated from the (resampled) output particle population. Thus, estimated covariances indicating uncertainty ellipses above a certain threshold size can be considered unreliable.

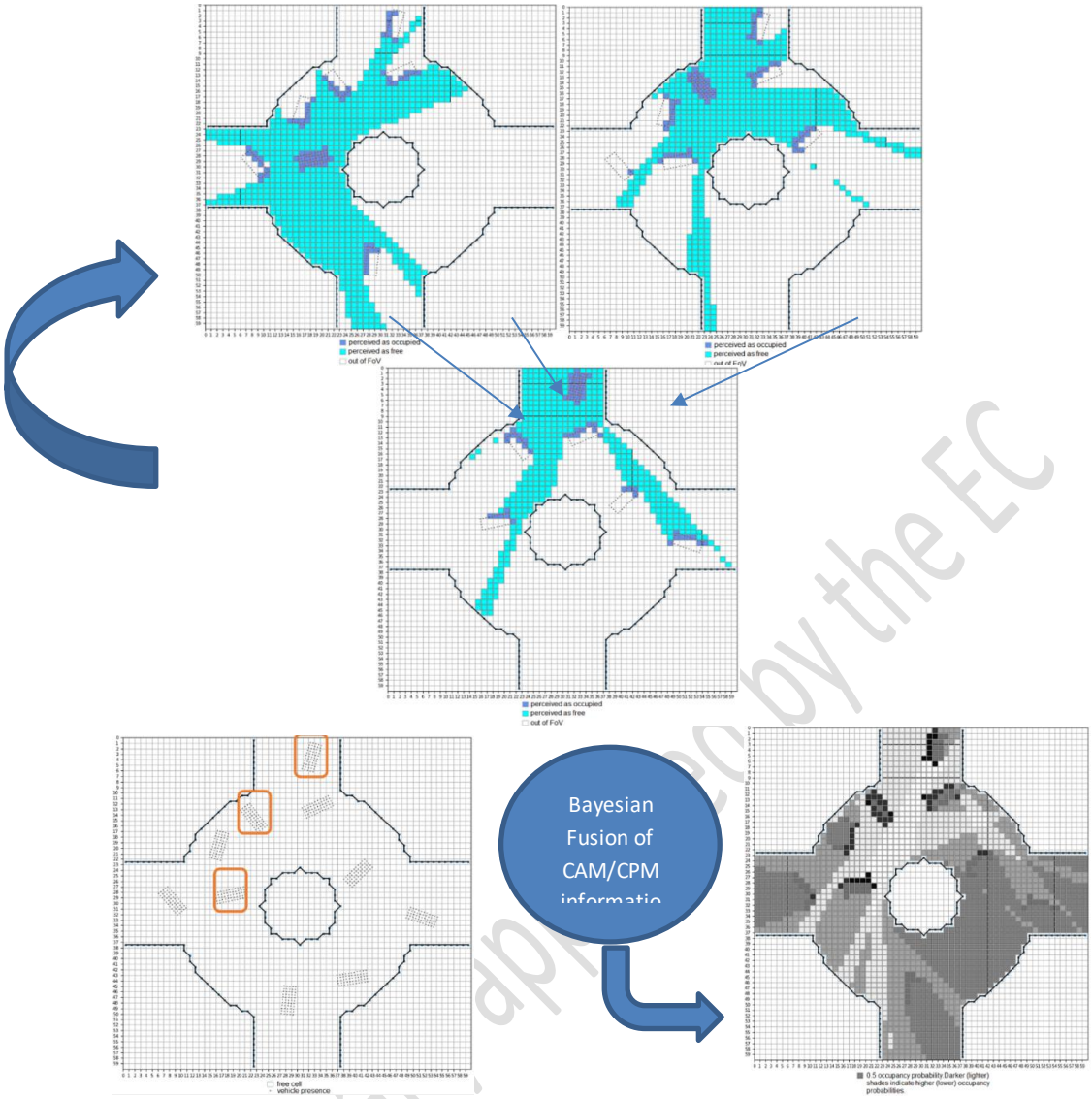


Figure 28: Scene BEV (left), CAV Field of View calculation (upper row images) and Fused Probabilistic Occupancy grid (right).

The FoV of each particular CAV is calculated based on its estimated location plus the locations of its perceived objects/obstacles, essentially classifying grid cells into three classes: visible and empty, visible and occupied, and invisible (i.e. outside the FoV). This information provides a basis for performing plausibility checks and assessing the consistency of the disseminated individual perception data [72], [73], [74], [75]. We distinguish two categories of such checks. (a) Checks detecting self-contradicting information. This essentially means that the objects claimed to be perceived by the CAV should be within the reporting CAV's FoV. Otherwise, one or both of the object's and CAV's location are considered untrustworthy. (b) Checks detecting contradicting information across all individual perception data disseminated by the CAVs. Several checks are possible; (b1) Cells claimed to be occupied by an agent that are within the FoV of other agents, should also be perceived as occupied by the other agents. (b2)

Cells perceived by an agent as occupied (or free) that are within the FoV of other agents, should be perceived as such also by the other agents.

The output is a probabilistic occupancy grid, a notion dividing grid cells into those whose occupancy probability is adequately high/low and to those it is neither. By definition, cells of the second class indicate the locations of the grid for which the occupancy estimates are unreliable. This may be due to missing information (e.g. cells may be invisible by every agent) or conflicts between individual perception data of the involved agents as described in the previous section. We emphasize that in each case, either lack or conflict of available information will be reflected in the Bayesian fusion and its resulting posterior occupancy probabilities.

Ultimately, there may be cells for which the output is reliable, and cells for which it is not. Hence, the overall approach does not address the issue of reliability of perception only as a (set of) general metric(s), but also accounts for its *locality*, a crucial feature in automated driving applications. Unreliable occupancy estimates may in fact be irrelevant when the associated grid cells are located far from the involved actors; a single unreliable estimate may be of utmost importance when it is close to one of them. The adopted approach captures this concept of locality in perception reliability in a fully transparent and explainable way.

3.3 EXP3 (UULM): Self-assessment and reliability of perception data with complementary V2X data in complex urban environments

EXP3 focuses on demonstrating safe automated driving in urban settings characterized by occlusions, relying on a combination of onboard SA methods and V2X data. The onboard perception system assesses its reliability, feeding that information into the further system to improve safety under challenging conditions, such as sensor failure.

Deliverable D2.1 [82] outlines user and system requirements for relevant use cases and provides a detailed explanation of EXP3. Additionally, Deliverable D2.2 [83] provides a system architecture design, which is a subset of the project's overarching master architecture.

This architecture, shown in Figure 29, illustrates how data flows internally between various modules, along with the corresponding inputs and outputs. For effective SA, a reliable foundation must be established, involving processes like data pre-processing, object detection, and object tracking.

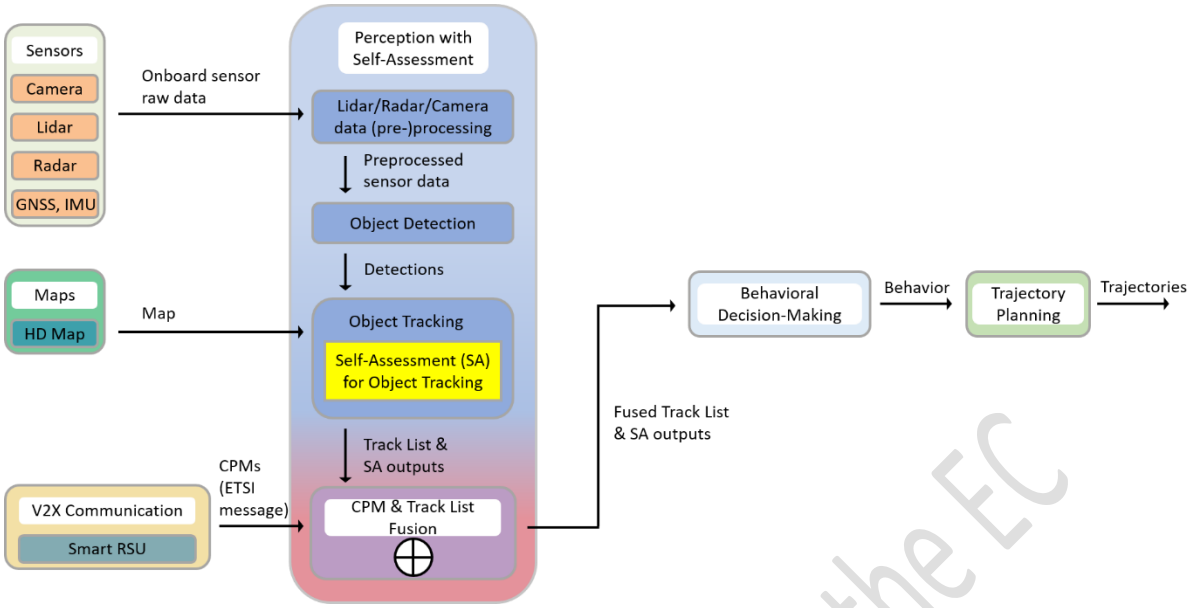


Figure 29: Overall architecture of EXP3 modified from [83]

In the scope of WP3 and this corresponding deliverable, the goal is to develop an SA framework for object-tracking algorithms such that the onboard perception system obtains SA scores. These scores are then used for the CPM and onboard track list fusion and the subsequent behavioural decision-making. First, an online performance assessment of multi-sensor Kalman filters is developed [88], and second, an SA framework for multi-object tracking is proposed [87]. These two approaches are based on the subjective logic theory [101], which enables a suitable framework for SA. The presented approaches continue the research of [90], [85], [91] toward safety and robustness in object tracking and, thus, autonomous driving. In addition to these works, more methods have been developed in the scope of WP3 and this project, namely [92], [94], [93], [95], [96], [97] .

3.3.1 Online Performance Assessment of Multi-Sensor Kalman Filters Based on Subjective Logic

Dynamic state estimation, particularly through the Kalman filter, is a key technique for filtering and tracking. Traditional methods like the normalized innovation squared (NIS) and normalized estimation error squared (NEES) [76] check specific criteria but provide limited insight into overall reliability. With increasing focus on reliability in the automotive industry, especially under ISO 21448 (SOTIF) [100], there is a need for a holistic SA framework.

This work introduces a novel online SA framework for linear and nonlinear Kalman filtering, leveraging subjective logic. Moreover, it extends the previous SA methods [90], [85] to nonlinear filtering and proposes a multi-sensor assessment framework that allows real-time performance evaluation without requiring ground truth data. The approach enhances reliability in multi-sensor systems, addressing the gaps in

current methods. This work has been presented and scientifically published at a conference in [88].

Framework

A novel aspect of the proposed system is its multi-sensor overall assessment framework, which directly integrates the single-sensor SA modules. This integration facilitates the assessment of multi-sensor configurations by leveraging the capabilities developed for individual sensors, providing a scalable and flexible solution for multi-sensor systems. The proposed framework is conceptually visualized in Figure 30.

Furthermore, the framework enables the derivation of closed-form solutions through the use of subjective logic. An additional key feature of the framework is its real-time capability. It is designed to continuously monitor and evaluate both the individual performance of each sensor and the overall performance of the filtering process. This real-time evaluation is crucial for applications that demand high levels of accuracy and reliability, such as autonomous driving and other safety-critical systems.

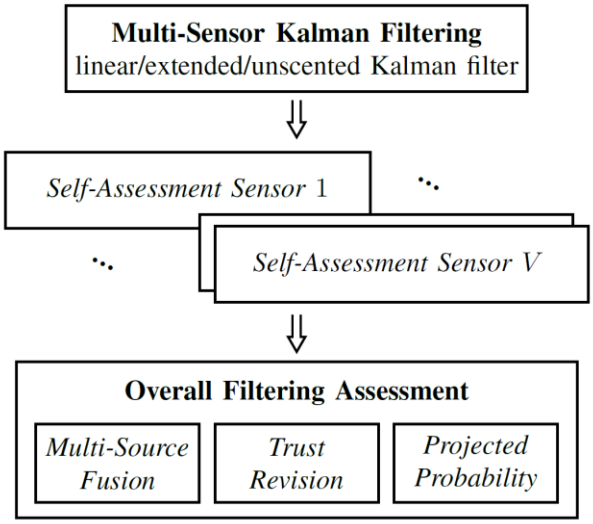


Figure 30: The SA framework for linear and nonlinear multi-sensor Kalman filtering from [88].

Simulation Results

In the simulation results shown in Figure 31, the three multi-sensor Kalman filter assessment approaches (multi-source fusion, trust revision, and projected probability concepts from subjective logic) are tested with five sensors tracking a single object. Sensor noise is disturbed at different times, with all sensors affected between time steps 300-600, then only four between 900-1200, and gradually fewer until only one is disturbed by step 3000. The subjective logic-based approach with trust revision performs similarly to the time-averaged NEES but does not report warnings when only one sensor is disturbed. The proposed method provides a reliability score between 0 and 1, detecting disturbances without ground truth data, unlike NEES.

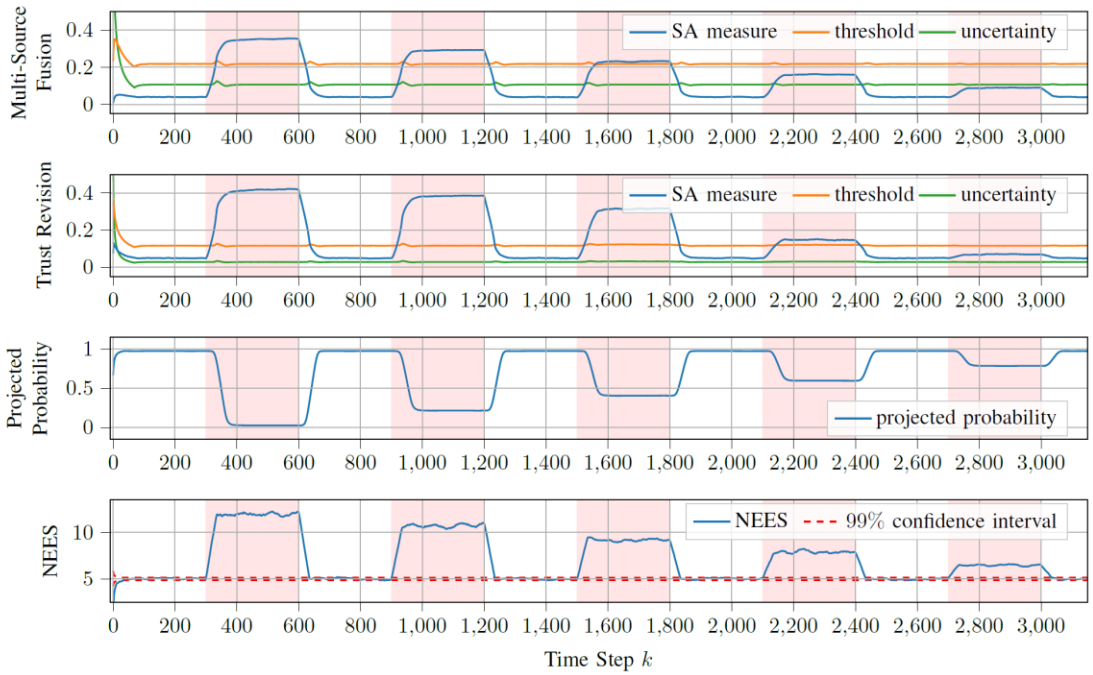


Figure 31: Simulation results of the proposed multi-sensor Kalman filter SA in a nonlinear scenario with five sensors from [88]. Measurement noise is disturbed for all sensors, then gradually reduced to one sensor (red areas). The subjective logic-based approaches (multi-source fusion, trust revision, and projected probability) are compared to time-averaged NEES using 200 Monte Carlo runs.

3.3.2 Self-Assessment for Multi-Object Tracking Based on Subjective Logic

As automated driving systems grow more complex, safety and robustness become critical challenges. To meet the ISO 21448 SOTIF standard, automated systems require SA modules, such as for multi-object tracking (MOT). Current methods, like the NIS, focus on single criteria, lacking a holistic SA approach for MOT.

This work introduces a comprehensive SA framework for MOT, ensuring tracking assumptions are monitored and validated. It presents a specific implementation using the global nearest neighbor (GNN) algorithm and subjective logic. The SA module is tested on real-world data from the KITTI dataset [84], demonstrating its practical use and contribution to safety and robustness in automated driving. The concept is visualized in Figure 32. This work has been presented and scientifically published at a conference in [87].

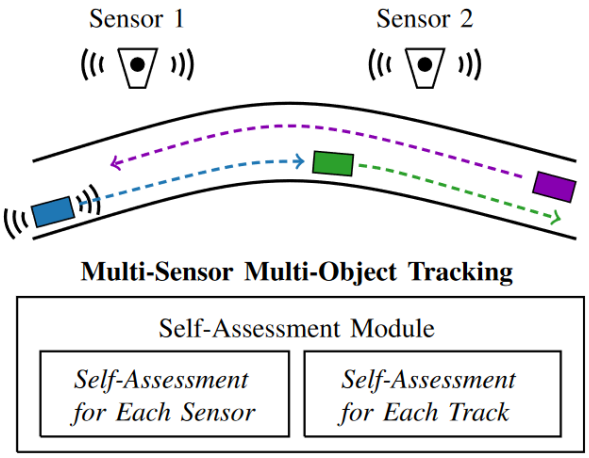


Figure 32: The proposed comprehensive SA module for multi-sensor multi-object tracking from [87].

Framework

This framework presents a comprehensive SA module for multi-sensor MOT, designed to calculate SA scores for each sensor and track. The implemented SA module includes two key components: the SA Sensor, which evaluates specific assumptions for individual sensors, and the SA Post, which assesses the algorithm's overall assumption fulfillment for each track. This is visualized in Figure 33.

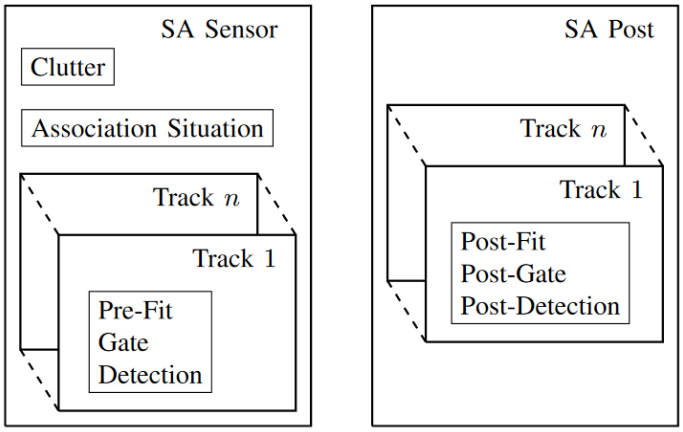


Figure 33: The conceptual overview of the SA module for multi-object tracking from [87]. The SA module, consisting of the SA sensor part and the SA post part, monitors tracking assumptions about clutter, the data association situation, pre-fit and post-fit residuals, the noise assumption within the gate, and the detection probability.

Experimental Results

The proposed SA module is applied to real-world scenarios using the KITTI 3D tracking dataset [84], focusing on the GNN tracking algorithm with lidar data. It is then evaluated on KITTI training sequences 4 and 10. The evaluation focuses on three specific aspects described below.

1. **Association Situation:** In sequence 4, many parked cars create ambiguous associations, which are reflected in a lower SA measure. Sequence 10, however, is a clear scenario that shows consistently high SA scores. The results are visualized in Figure 34.

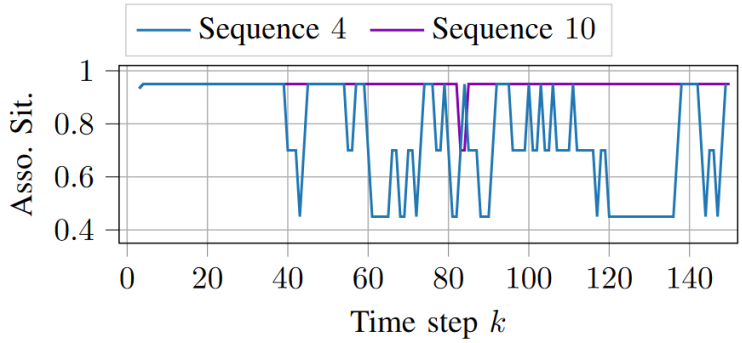


Figure 34: Association situation of sequences 4 and 10 of the KITTI dataset from [87]

2. **SA Module:** For sequence 10 in Figure 35, the subjective logic-based SA measures confirm that tracking assumptions are met, with a leading track detected in every time step. The MNIS noted some increases and violations across all tracks, but clutter SA measures also indicated that assumptions are satisfied.

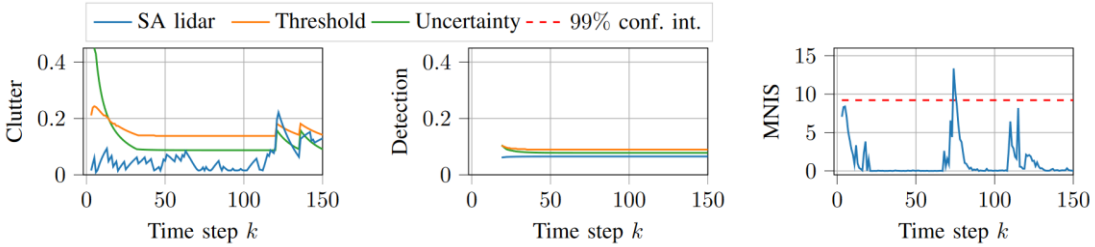


Figure 35: SA measures from the SA Sensor module for KITTI sequence 10 from [87], showing clutter and detection opinions compared to the multi-target NIS (MNIS). The detection SA measure focuses on the leading vehicle track initiated at time step 18, while the MNIS includes all tracks.

3. **False-Positive Tracks:** Sequence 10 also revealed false-positive tracks initiated by consecutive clutter measurements in Figure 36. The SA module effectively distinguished these from true object tracks, as evidenced by high detection uncertainty prior to deletion, which could enhance the reliability of track maintenance algorithms.

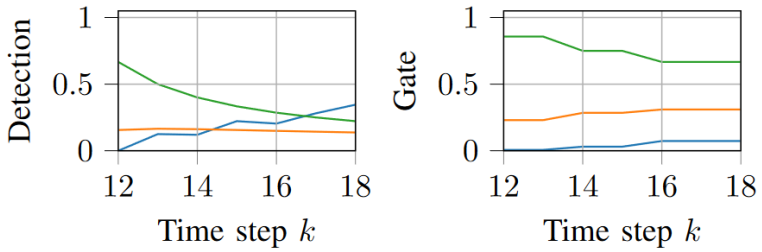


Figure 36: SA measures of a false-positive track in sequence 10 of the KITTI dataset from [87]. SA module successfully identifies the false-positive track prior to the deletion algorithm's response.

3.4 EXP4 (HIT, TECN): Decision making for motion planning when faced with roadworks, unmarked lanes and narrow roads with assistance from perception self-assessment

3.4.1 Introduction

The objective in EXP4 is to perceive and control a vehicle in the context of a unstructured road use case (specifically road works). The experiment first revolves around perceiving the environment to update the high-definition map. This information is then used by the motion planning module to follow a safe trajectory.

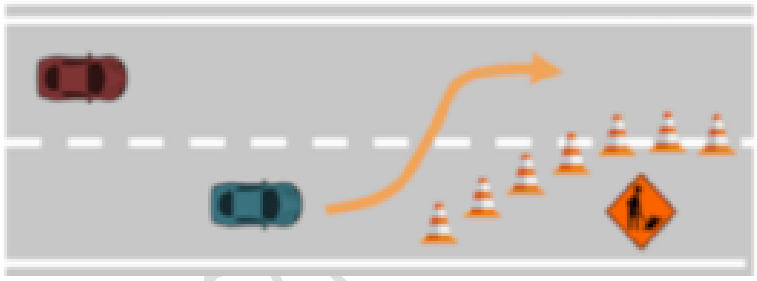


Figure 37: The ODD being considered in experiment 4. Given a two lane road, a single lane is blocked by traffic bollards. The lane structure is thus modified according to the position of the bollards.

3.4.2 HD-Map Update Using Detected Bollards

The proposed system seeks to update the HD-map in real-time based on the detected road work bollards in the scene. By updating the HD-map in real-time, we propose to enable more efficient motion planning that can constrain more effectively the candidate trajectories. HD-map generation is an active area of both research and commercial deployment. Examples of such systems exist in both real-world deployments [114] to research work that utilize multiple sensory sources and machine learning to generate an HD-map [115], [116]. While such works focus on HD-map generation, other works have considered updating HD-maps according to changes observed between the reference map and the scene as observed by the vehicle's sensors (our work falls into this category). A summary of existing methods can be

found in [117] with no specific roadworks use being mentioned (to the best of our knowledge).

To this end, we propose specifically to update the HD-map according to the observed roadworks bollards (using data obtained only from camera and GPS). Our approach will assume a specific ODD, namely, a 2 lanes road with one lane being blocked by road works. Our workflow is shown in **Error! Reference source not found.** and is c omposed of the following steps: 1) 2D detection of road work bollards, 2) 2D->3D estimation of the road work bollards, 3) generation of plausible lane boundaries and 4) update of HD-map based on the plausible lane boundary.

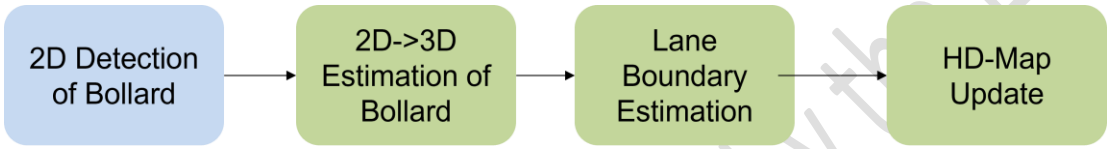


Figure 38: Workflow for detection of bollards to update HD-map

In the following subsections, we provide a description of the methods we have implemented.

3.4.3 2D Object Detection of Bollards

Our 2D object detector has been trained using both the proposed patch augmented data for traffic signs and bollards along with a subset of the Zenseact dataset that only includes road work bollards. We re-trained YOLOv5 to detect the following classes: 1) 20 speed limit, 30 speed limit, 50 speed limit, 70 speed limit, Road work, and bollards (we have limited the example of bollards to the one shown in Figure 39, however, cones are all used but we do not currently consider those for the EVENTS project). An example of our model detecting bollards on data that we have collected is shown in Figure 39.

Figure 39: Trained 2D object detector. The bounding boxes (in purple), shows our model correctly predicting the location of the road work bollards in the image. This image was captured from a HIT vehicle.

3.4.4 Estimation World Position

Estimating the world coordinates (position) of the detected bollards Is determined using a computationally efficient geometric approach. We chose this approach as the following criteria were satisfied, 1) the road work bollard in question is planar and aligned vertically with respect to the ground plane, 2) LiDAR synchronization with camera is difficult to achieve, and 3) monocular depth estimation methods based on deep learning are computationally expensive and use limited GPU memory.

Our geometric based approach is based on photogrammetry, where a distance in the real world can be associated to a distance within the image.

3.4.5 Generate Plausible Lane Boundary

After estimating the world position for a set of bollards, we proceed to link them in a structured manner to form a lane boundary. This can be formally described by treating the bollard positions as nodes in a graph and aiming to predict the corresponding edges that connect these nodes. In our current work, we construct a minimum spanning tree to link the nodes (bollards) in the graph.

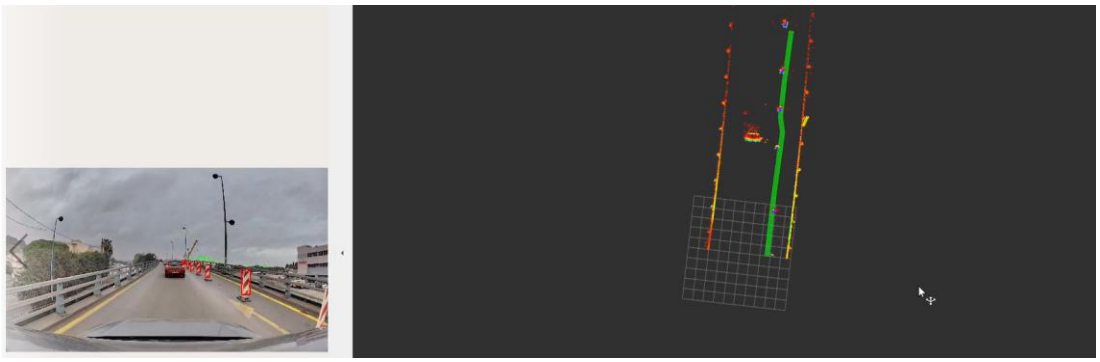


Figure 40: An example of the generated lane boundary (shown in right panel green line), after detecting the bollards (left panel red bounding boxes). The generated lane boundary is overlaid onto LiDAR data.

3.4.6 Update HD-Map

After obtaining the plausible lane boundary, the final step involves updating the HD-map. For this purpose, a rule-based process is used, which comprises the following steps:

1) Check if plausible lane boundaries are enclosed by the current lane boundaries derived from the existing HD-map, as shown in step 1 in Figure 41Figure 42, where the plausible lane boundary is enclosed by the left and right drivable boundaries.

2) If step 1 is true, determine if the plausible lane will form the "left" or "right" drivable road boundary of the ego vehicle's updated HD-map. This is shown in step 2 of Figure 41 by assessing the plausible lane boundary nodes with respect to the ego-vehicle. This processing step can be quite challenging for complex road configurations.

3) We finally update the HD-map by inserting the "new" boundary of the *drivable road.* This is shown in step 3 of Figure 41, where the drivable road boundary is updated based on the class of the plausible lane boundary determined in step 2.

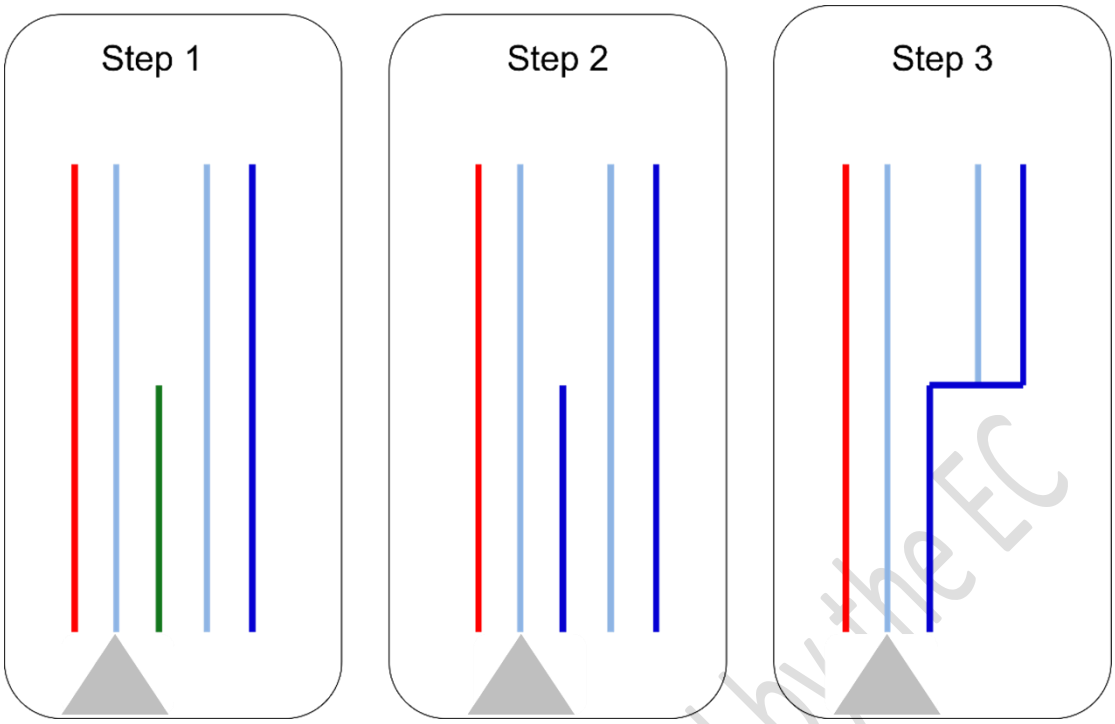


Figure 41: An illustration of the proposed approach for updating the HD-map based on the steps outlined in Section ***Error! Reference source not found.****. The red line corresponds to the " left" drivable road boundary, while the blue line corresponds to the right drivable road boundary. The green line is the plausible lane boundary estimated using the method described in Section* ***Error! Reference source not found.****.. The light blue lines are the lane c enterlines. The gray triangle represents the ego vehicle.*

Finally, the screen shot depicted in Figure 42 shows the full pipeline with data collected from HIT vehicle in a scenario that matches our ODD.

Figure 42: Final system shows the updated the HD-map. Example is shown in the map frame (against the original map frame HD-map shown by the green lines). The red lines on the right panel are the left boundary of the drivable road, and the blue line is the right boundary of the drivable road. The light blue lines are the centerlines of the lane/s.

3.5 EXP5 (HIT, TECN, WMG): Predictive perception when merging onto a highway

3.5.1 Introduction

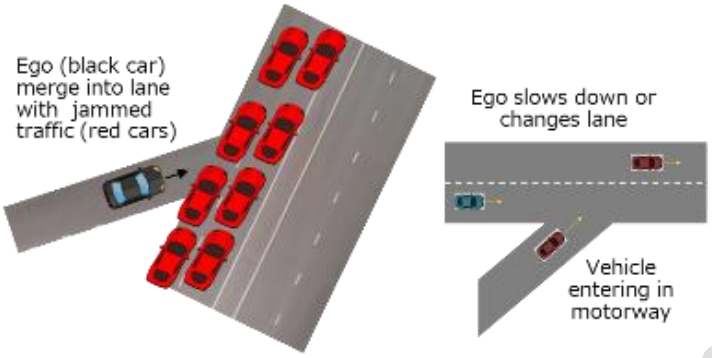


Figure 43: Experiment 5 scenarios.

In this experiment, HIT, TECN and WMG cooperated to tackle scenarios where the ego-vehicle attempts to merge onto a highway or is in a highway lane observing other vehicles merging onto the main road, as shown in Figure 43. To address these situations, the target system needs to quickly detect and track multiple moving objects at varying speeds and predict their future movements without relying on maps. To this end, we developed and contributed a predictive perception system capable of robustly detecting and tracking multiple 3D objects moving at both low and high speeds in real-time (HIT's contribution) and predicting their future movements based on past trajectories (TECN's contribution). In addition, WMG contributes to the perception SA of the system. This enables the ego-vehicle to quickly access reliable information for safe decision-making at intersections. These modules were developed and tested using data collected by Hitachi's demo vehicle.

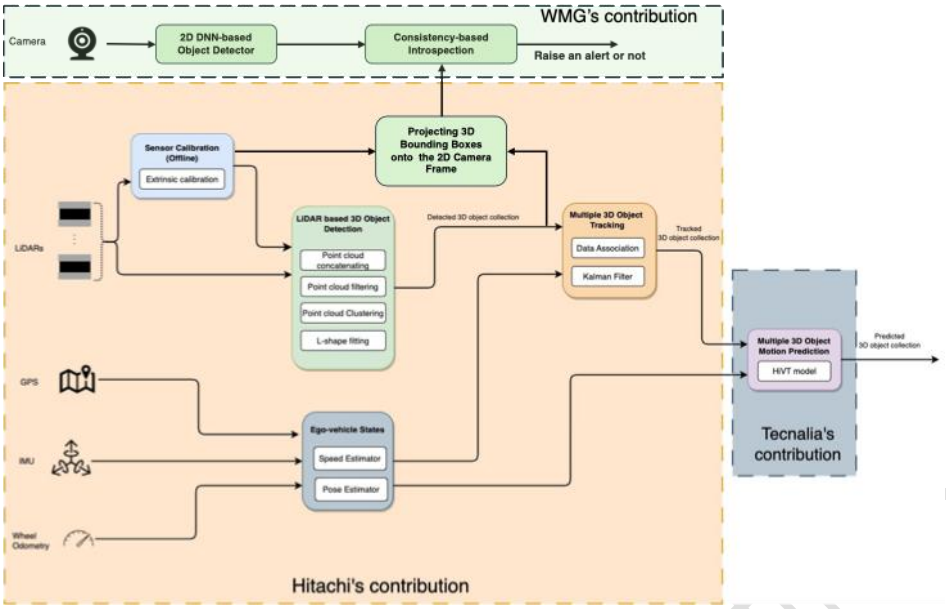


Figure 44: Software architecture for EXP5 with corresponding partner contributions

Figure 44 shows the software architecture for EXP5. HIT contributed to the end-to-end system from sensors to tracked 3D object collection (Figure 44 left block) including: sensors installation, sensors processing and calibration, LiDAR based 3D object detection, multiple 3D object tracking module. TECN contributed to the multi 3D object motion prediction module (Figure 44 right block). In this experiment, WMG contributed a perception monitoring mechanism that uses consistency checking between LiDAR and camera (Figure 44 top block).

3.5.2 Multiple 3D object detection and tracking

3D object detection focuses on estimating three-dimensional rotated bounding boxes using images or LiDAR data. It is an indispensable component of 3D multi-object tracking because the accuracy of these bounding boxes significantly impacts tracking performance. Compared to camera-based methods, LiDAR-based 3D object detection methods [119], [120], [121] deliver impressive results due to the precise 3D structural information provided by point clouds obtained from LiDAR sensors.

3D MOT involves detecting and continuously tracking multiple objects across frames in a video, maintaining their identities even as they move or change appearance and based on that the velocity of interested objects can be attained as well. LiDAR-based detectors are widely favored for 3D MOT [122], [123], [124] due to their simplicity and high effectiveness.

Due to the above reasons, as well as the experiment required a fast frame-rate 3D MOT output, HIT has contributed on developing a robust and real-time perception system that include a LiDAR-based 3D object detection and MOT algorithms to address the challenges.

As depicted in Figure 44, in the 3D object detection module, we developed a point cloud clustering algorithm to segment points into different clusters after concatenating and filtering points from two LiDARs. Those clusters are further processed by L-shape fitting algorithm to extract the 3D bounding boxes. Using this method, we were able to achieve 3D bounding boxes at a frame rate of 15 frames per second on the data collected by HIT's demo vehicle. An example processing output of the algorithm on Hitachi collected data is showed in Figure 45.

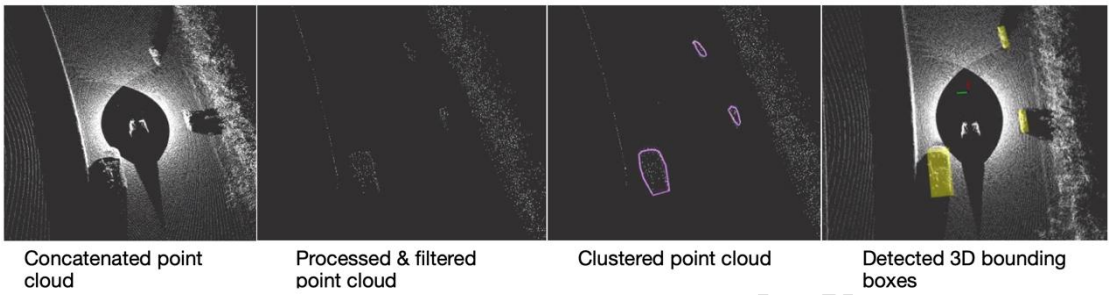


Figure 45: Example outputs from 3D object detection algorithm at an intersection of speed limit 70Km/h.

As shown in Figure 44, the output of the 3D object detection algorithm is fed into the 3D MOT algorithm. As explained in D3.1 [25], we developed a Kalman filter and a data association algorithm to track multiple 3D objects. We further improved the output of tracked object by providing the motion compensation from ego-vehicle's state as well as a data association that can associate object in different frames when it moves fast. As a result, our 3D MOT algorithm can track multiple objects with a high range of relative speed with respect to the ego-vehicle from 0 km/h to 150 km/h. The 3D MOT algorithm processes the data collected by Hitachi's demo vehicle at a rate of 15 frames per second. An example of the tracking output at a high-speed road intersection can be seen in Figure 46.

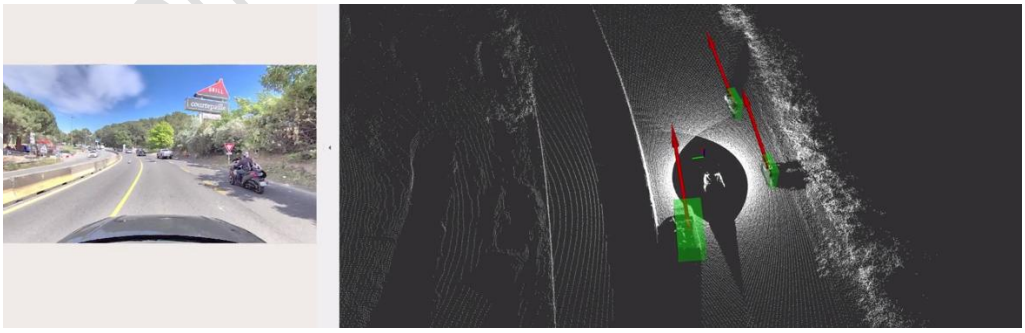


Figure 46: Example tracking output on Hitachi's collected data. Left: Frontal camera image, Right: Tracked 3D object (in green) with indicated velocity (red arrow).

3.5.3 Motion prediction

Motion prediction plays a crucial role when merging onto a motorway. The vehicle must calculate the future trajectories of the other vehicles in order to merge safely and efficiently.

For this reason, TECN has trained HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction [118] without the map information, for scalability to other setups where the map is not available as it is in Experiment 5.

The motion prediction module needs the history of the surrounding vehicles. They must therefore be accurately detected and tracked. Hitachi provides this information, which is fed into the module. The objects are then transformed into the ego-vehicle reference to infer the relationships between the agents in the scene. These trackers are organized as a collection of vectorized entities. The model encodes their social interactions to compute their future trajectories. The model predicts six trajectories per agent, representing the next 3 seconds at 10 Hz, as described in Experiment 2. However, in this case the information is only for an equipped vehicle rather than a collaborative maneuver.

3.5.4 Perception system self-assessment

In complex driving scenarios such as merging areas, an accurate detection of objects can be challenging. To improve safety, an SA mechanism can be introduced to continuously monitor and verify the quality of the primary perception system and raise an alert when needed. This section presents the SA framework of EXP5 through monitoring inconsistencies between the primary perception system, i.e., a clustering-based 3D object detector, and an introduced camera-based 2D object detector.

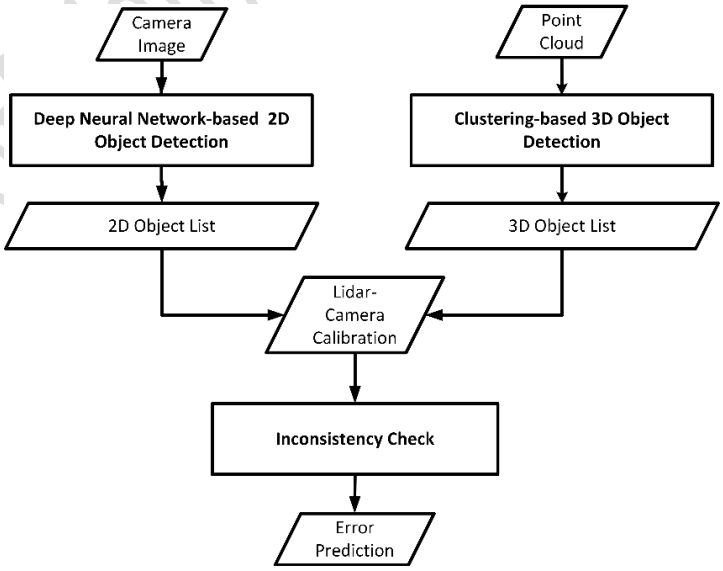


Figure 47: SA framework using 2D and 3D object detection with an inconsistency check to predict errors from LiDAR and camera data.

As illustrated in Figure 47, the SA framework comprises three main components: A 3D object detection module that uses clustering, a SOTA DNN-based 2D object detection module and the Inconsistency Check as a SA mechanism. The DNN-based 2D Object Detector processes camera images to generate a list of 2D detected objects. Simultaneously, the clustering-based 3D Object Detection module works on LiDAR point cloud data to generate a list of 3D detected objects. These two results are aligned through LiDAR-Camera Calibration, ensuring that objects from the 2D and 3D domains are mapped correctly. The Inconsistency Check module then compares the outputs of the 2D and 3D object lists to identify any potential discrepancies. Specifically, an inconsistency is flagged when an object detected by the camera with a confidence score greater than e.g., 0.5, does not have a 2D bounding box that achieves an IOU higher than e.g., 0.4, with any of the 3D bounding boxes (after projecting them onto 2D) detected by LiDAR-based clustering. If an inconsistency is detected within the input frame, an alert should be raised. Apparently, there's a safety concern, if both perception mechanisms miss an object. In that case, more sophisticated SA mechanisms such as those developed for EXP7 can help.

The inconsistency-based SA mechanism in EXP5 is further clarified using the example illustrations of Figure 48. For the input frames depicted in Figure 48a, 48b and 48d, either the IOU between the bounding boxes generated by camera and clustering method exceeds 0.4, or the camera has failed to detect some objects in the scene, likely due to occlusions. In contrast, for the input frame shown in Figure 48c, the clustering algorithm has missed the white car on the left-hand side, which may pose a safety risk due to its potential conflict trajectory with the ego-vehicle. The camera, however, has detected this car with a confidence score greater than 0.5. Therefore, the SA mechanism raises an alert in this case and thereby enhancing safety.

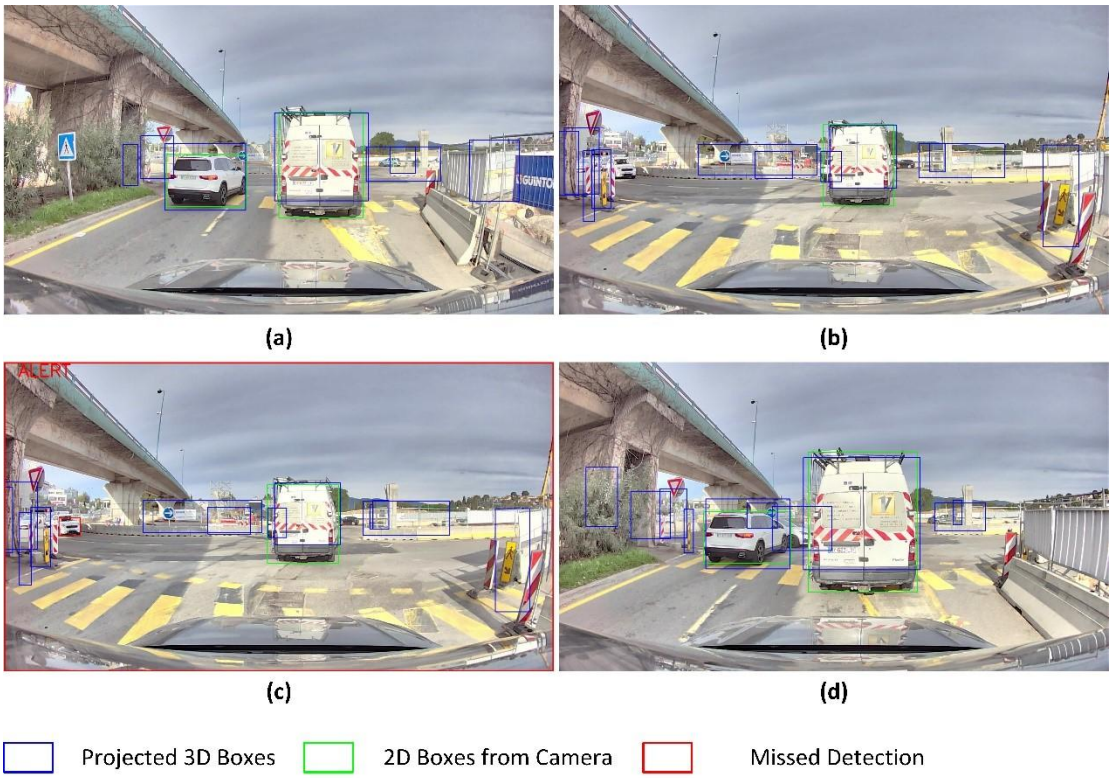


Figure 48: Example illustrations of the SA mechanism in EXP5.

3.6 EXP6 (APTIV): Small object detection at a far range in adverse weather conditions

3.6.1 Introduction

The objective of EXP 6 is to develop a perception system able to detect small objects in difficult weather conditions. A radar was selected as the base of the perception system as vision systems can be impaired in low visibility conditions [125]. Radars can provide more accurate information on the position in longitudinal direction of the observed objects and they can also provide the range rate of reflected surface. The radar selected for the experiment provides an elevation angle of detected reflection.

The detected object has to be classified as either overdriveable or non-driveable in order to determine the appropriate braking response. For safe and uninterrupted driving, the system should only slow down for overdriveable obstacles like speedbumps and small debris without coming to a complete stop. In contrast, objects that could potentially cause damage upon collision should be considered as non-drivable and trigger a deceleration response brining the vehicle to complete stop before the obstacle.

The detection of debris as part of Automated Driving Assistance Systems (ADAS) or autonomous driving systems can improve safety. According to [126], between 2011 and 2014, an estimated 50,658 debris-related accidents occurred annually in the

United States, resulting in an average of 9,805 injuries and 125 deaths each year. A more detailed analysis of accidents involving debris on the road can be found in **Error! Reference source not found.**.

For a description of the newly recorded road debris dataset, refer to Section 2.

3.6.2 Overview of perception

Radar data contains azimuth and elevation angles, range, range rate of reflection points in the Vehicle Coordinate System (VCS), and the radar cross section of the reflections. Each data point from a single radar scan will be referred to as a radar detection. Radar detections are clustered using the DBSCAN algorithm [127]. The resulting clusters of radar data are provided to an algorithm which creates polylines from stationary objects. A polyline is a connected sequence of line segments on a 2D plane, which is a representation of stationary environment (objects that are not expected to ever move); refer to ISO 23150:2023 [128] for a more detailed definition. Most of the objects detected by the front-facing radar during one cycle of measurement provided a singular point of reflection. Radar detections associated to a polyline are accumulated over time. The positions of historical detections are updated in the VCS based on the host vehicle's speed and yaw angle. The perception stack for processing these detections is shown in Figure 49.

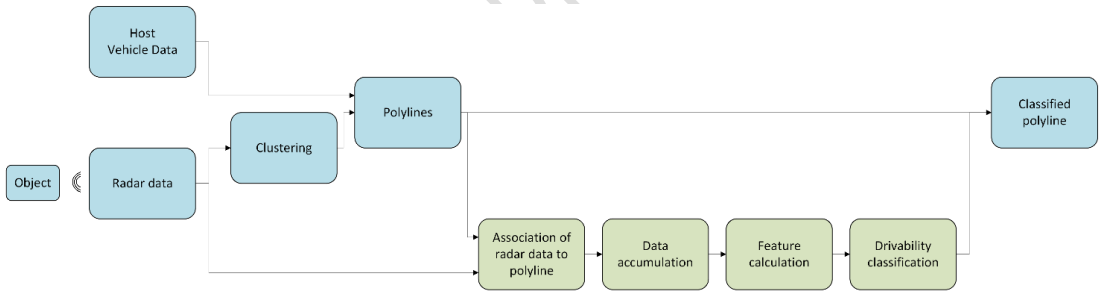


Figure 49: Perception data pipeline.

3.6.3 Classifier training

Naive height discrimination between different classes of objects using the elevations of detections from a radar scan, would not produce any acceptable performance. To address this, a machine learning-based classifier has been designed. More than 20 features were selected through expert's judgement as input to the classifier. These features were derived from raw detections properties. Features are calculated for a polyline based on aggregated data from radar detections associated to polyline.

Ground truth classification labels are based on the height of an object – objects smaller than or equal to 12 cm are considered as overdriveable (refer to [25] for our reasoning

to use this threshold value). Manual labels, used as ground-truth, are associated to polylines created on the observed object.

Radar detection data, features calculated using that data, and labels are used in the machine learning process to train an appropriate classifier. Examples of methods of classification via machine learning using radar data are discussed in [129], [130]. The data collection and algorithm training pipeline is shown in Figure 50.

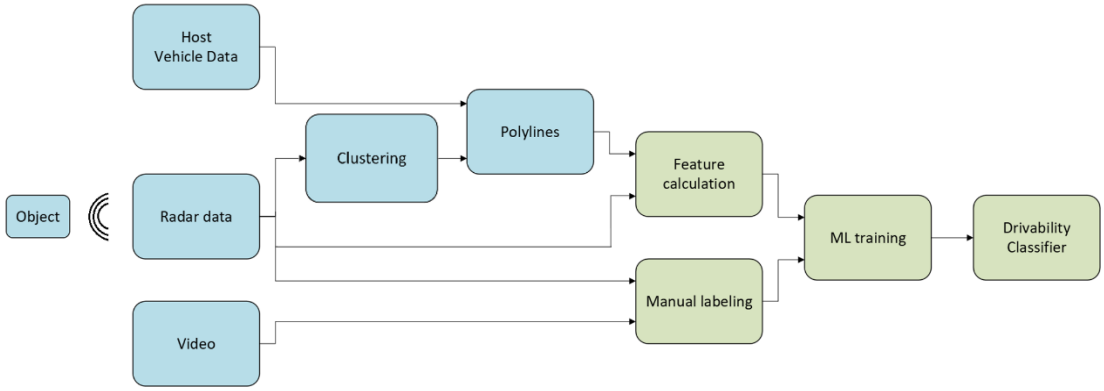


Figure 50: Process of data preparation for training overdriveability classifier.

3.6.4 Data collection measurements overview

Figure 51 shows some of the objects used to test the overdriveability classifier. The brick is considered as overdriveable and the rest are non-driveable.

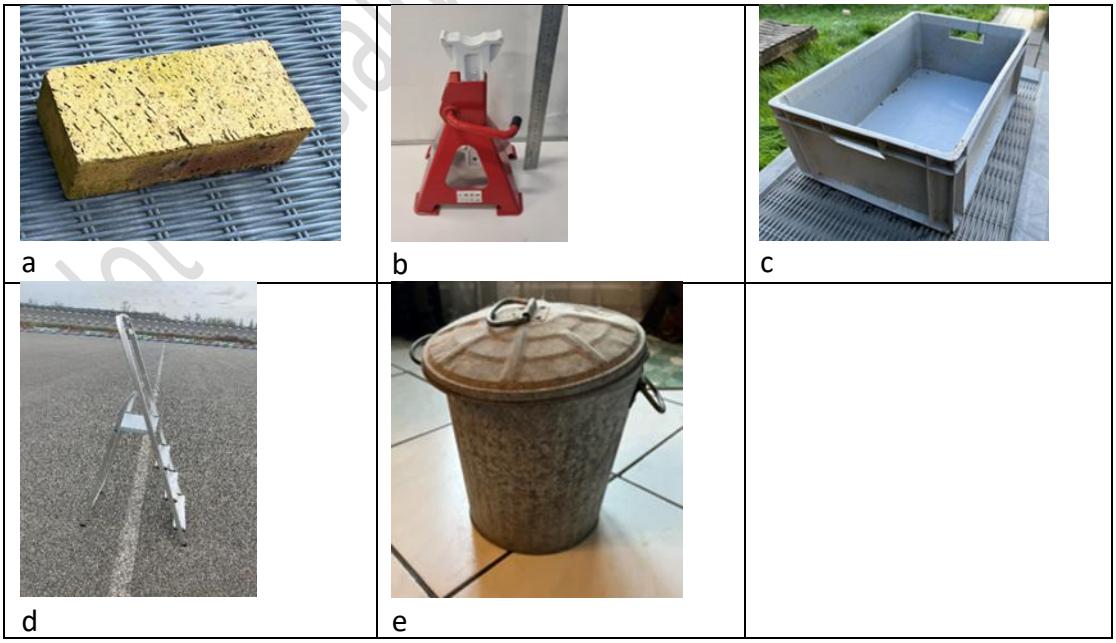


Figure 51: Object under test – (a) brick; (b) axle stand; (c) box; (d) standing ladder; (e) metal bucket.

As expected, such small objects provide a single point of detection most of the time as seen in Figure 52.

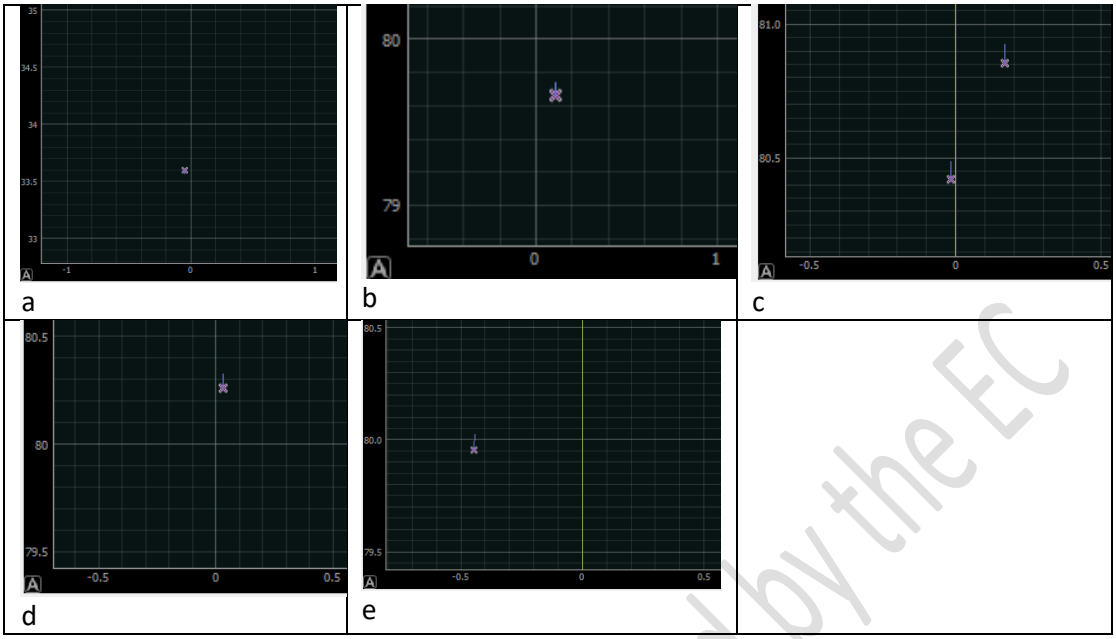


Figure 52: Radar data from one scan. (a) brick; (b) axle stand; (c) box; (d) standing ladder; (e) metal bucket. The axes show the longitudinal and lateral distance to the host vehicle in metres.

3.6.5 Classifier output results

Radar data gathered on the test track was used to train an overdriveability classifier. Accumulated detections used for feature calculation are presented in Figure 53.

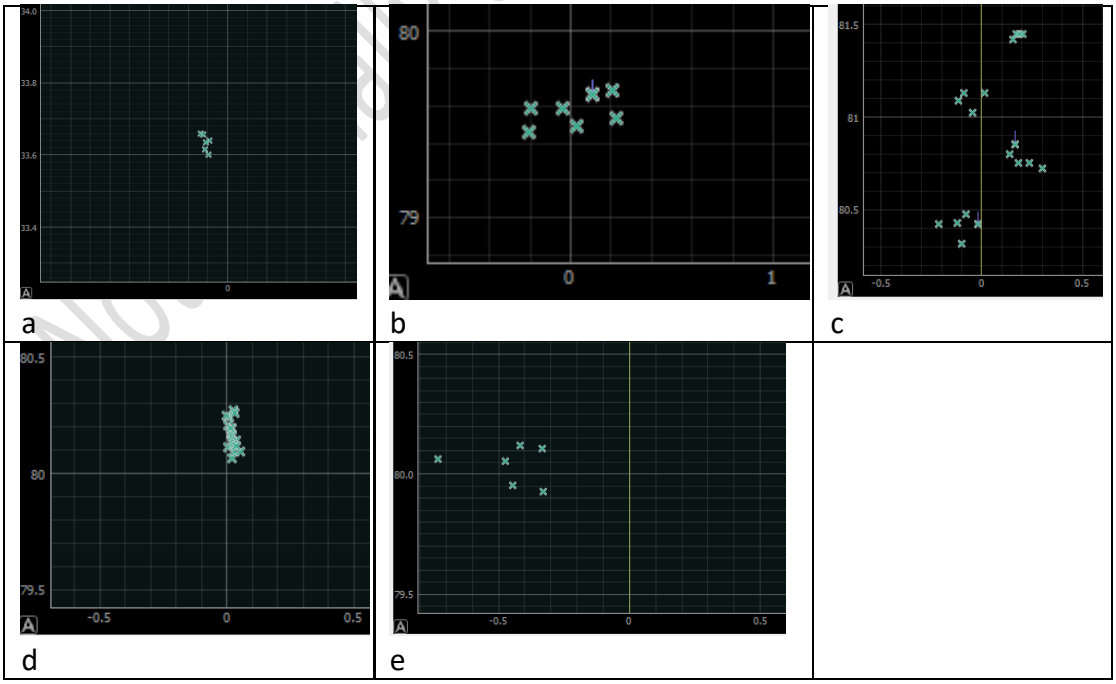


Figure 53: Accumulated radar data. (a) brick; (b) axle stand; (c) box; (d) standing ladder; (e) metal bucket. The axes show the longitudinal and lateral distance to the host vehicle in meters.

The output of the algorithm is a polyline with overdriveability classification. Polylines created on a small object will contain only two vertices and one segment. The classification results for the five objects shown in Figure 51 are presented in Figure 54.

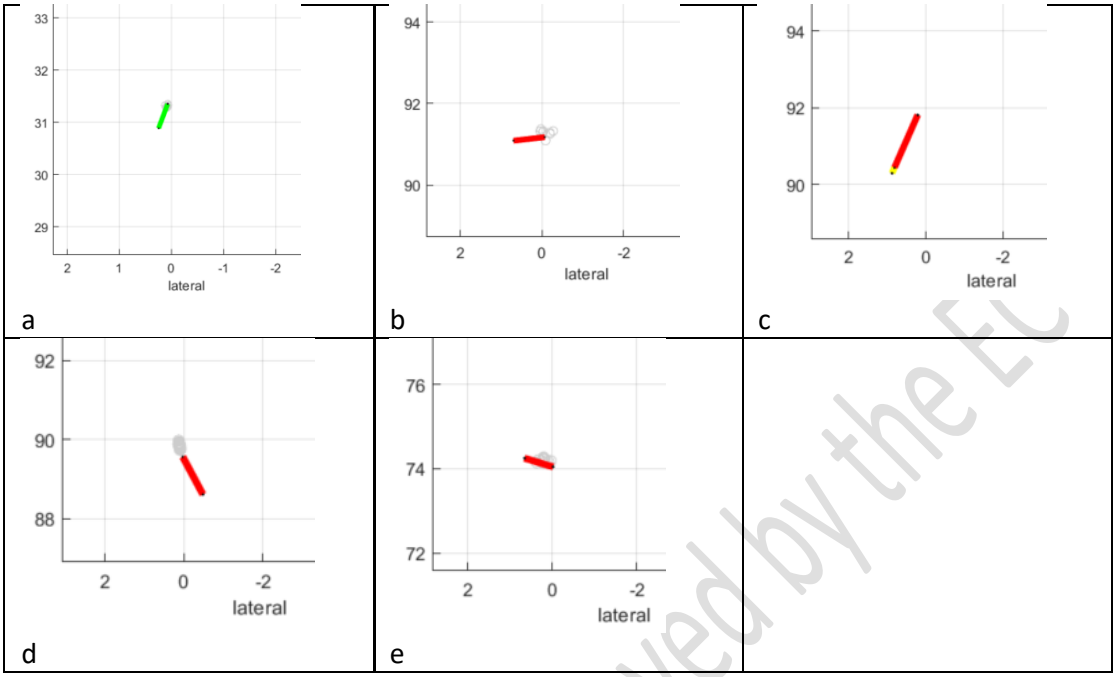


Figure 54: Polyline with overdriveabilty classification: (a) brick; (b) axle stand; (c) box; (d) standing ladder; (e) metal bucket. The axes show the longitudinal and lateral distance to the host vehicle in metres. Red means non-driveable and green means overdriveable.

3.7 EXP7 (ICCS, WMG): Localization/perception self-assessment for advanced ACC and other vehicles' behavior prediction under adverse weather or adverse road

EXP7 is titled as "Localization/perception SA for advanced Adaptive Cruise Control (ACC) and other vehicles' behaviour prediction under adverse weather or adverse road conditions". In WP3, the objective is to develop SA mechanisms for (i) LiDAR-based 3D object detection, and (ii) LiDAR-based localisation with respect to the leading vehicle for ACC. This is motivated from the well-known limitations that adverse weather poses on the LiDAR's object detection performance, as well as the challenges of distance estimation in urban settings with curved road segments. In both cases, we will implement an actor-critic architecture for SA, where the actor represents the perception system, and the critic acts as a secondary system that continuously monitors and assesses the performance of the actor. Once the critic detects a performance degradation in the actor, it is supposed to trigger a handover request to the human driver in SAE L3, or a safe minimum risk manoeuvre in SAE L4 autonomy. The development of perception SA mechanisms is therefore crucial for enhancing the overall safety and trustworthiness of Automated Driving Systems (ADS).

3.7.1 Self-assessment of LIDAR-based 3D Object Detection

Perception systems are responsible for accurately detecting, classifying and tracking road users in the vehicle's surroundings, serving as the foundation for the subsequent motion planning and control processes. Despite recent advancements, SOTA DNN-based detectors, which are primarily employed for object detection tasks, remain susceptible to errors [131]. Because of that, various methods for the SA (or introspection) of DNN-based 2D/3D object detection systems have recently emerged, leveraging distinct input representations and techniques for identifying object detection errors [132].

Existing introspection methods often rely on the confidence level of the object detector, as discussed in [133]. However, DNNs are known to perform poorly in estimating the uncertainty in their predictions, often being overconfident. Some studies aim to provide more accurate confidence scores within the main model by using methods such as sampling [134], confidence calibration [135], or confidence estimation [136]. Another approach for introspection involves detecting discrepancies (inconsistencies) through different systems running in parallel, exemplified by the comparison of the outputs provided by object detection and tracking in [137]. In controlled environments, a history-based SA technique has shown some promises, as investigated in [138]. Lastly, predicting when the detection performance, indicated by metrics like the mean average precision (mAP), would drop below a specific threshold can help in pinpointing errors, as detailed in [139], [140], and [141] for camera-based 2D object detection.

In the following, we describe a framework that leverages multi-layer neural activation patterns combined with spatial filtering to provide a detailed and dynamic SA mechanism for LiDAR-based 3D object detection. The SA framework is specifically trained to raise an alert when a vehicle or pedestrian is present but not detected within a defined area of interest around the ego-vehicle.

Framework description

The proposed framework uses neural activation patterns across multiple layers, combined with spatial filtering to focus on critical areas around the ego-vehicle. The SA pipeline, indicated by the red arrows in Figure 55, is detailed below.

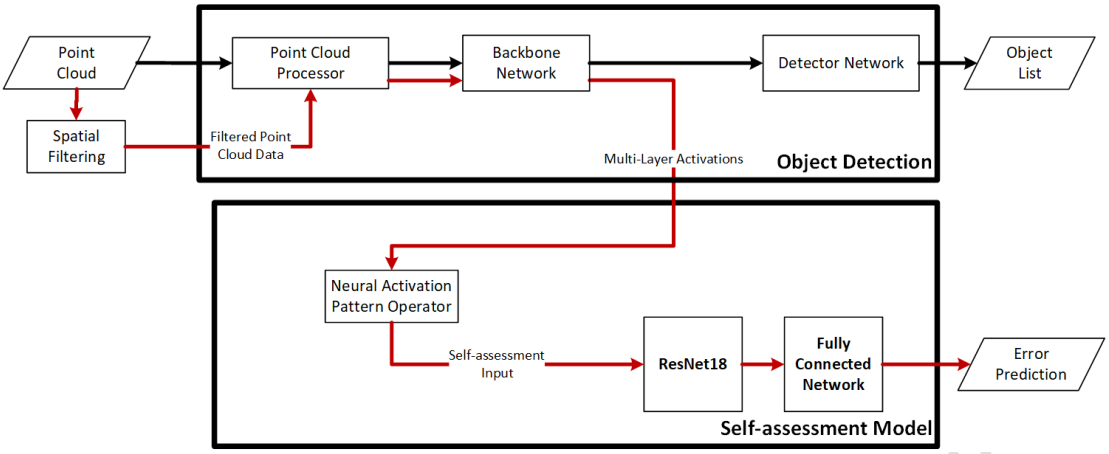


Figure 55: Object detection and SA pipelines during inference. Black arrows indicate the flow of information for the object detector and red arrows for SA.

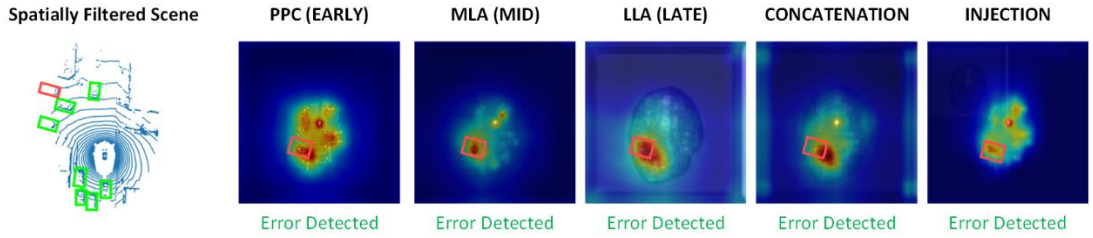
Initially, neural activation patterns are extracted from various layers of the Backbone Network within the Object Detection model. These patterns are processed by a Neural Activation Pattern Operator, which applies custom preprocessing techniques or selectively use patterns from different parts of the Backbone Network. The processed activation patterns are subsequently fed into a ResNet18 model, which extracts features essential for SA. Lastly, a Fully Connected Network analyses these features to make an Error Prediction (binary output), determining whether the 3D object detector has accurately detected all relevant objects, or if potential errors exist. It is noted that the SA framework includes a Spatial Filtering mechanism to guide the model to focus only on the vicinity of the ego-vehicle. Although this is optional, it is naturally more practical to concentrate on the immediate surroundings for missed objects as opposed to creating alerts for missed objects that are far away.

Qualitative evaluation example

The performance of the developed mechanism is evaluated using the NuScenes dataset, which has been widely used in both introspection and ADS domains. For LiDAR-based 3D object detection, we employ the CenterPoint model, which is implemented in the Autoware Foundation's software stack [142], while for the introspection model, various configurations are implemented, and their performances are compared. Specifically, we investigate the performance with five different settings of the Neural Activation Pattern Operator, refer to Figure 55.

In the first three settings, we have used activation patterns from individual layers as follows: Processed point clouds (PPC), middle layer activations (MLA) and last layer activations (LLA). For the remaining two settings, we have combined neural activation patterns from multiple layers through two different combination approaches. The first approach, called CONCATENATION, simply concatenates multiple activations, i.e., PPC, MLA and LLA, and feeds the combined activations into the ResNet18 network. To

concatenate different activation maps, down-sampling might be needed to resolve any resolution mismatch. The second approach, called INJECTION, injects different activation patterns at different stages of the ResNet18 network to preserve information and provide robust SA output.



(a) (b)

Figure 56: Spatially filtered point cloud. The detected objects (rectangles) are coloured in green while the missed object is coloured in red. The driving direction is from bottom to top (a). Early layer activation maps for each setting of the Neural Activation Pattern Operator illustrate the focus area of the SA model. Red hues represent high focus while blue hues indicate low focus (b).

To better understand how the SA models perceive the scene across the five settings, we have used an input frame and visualised the areas of focus through activation maps from the early layers of ResNet18. We have generated these visualisations using Ablation-CAM [144], a well-known method for activation visualisation that removes individual feature maps and measures their impact on the prediction. For a more comprehensive visualization, we have extracted activation maps from all residual blocks.

Figure 56 presents the input scene after spatial filtering at the left-hand side, alongside visualisations of activation maps for each of the five settings of Neural Activation Pattern Operator. The scene representation using point cloud includes green boxes for correctly detected objects and red boxes for missed detections, showcasing distinct patterns of critical areas used for classification. The visualisations highlight that while all five settings detect a missed object in the area of interest ("Error Detected"), the INJECTION method provides more focused attention on the missed object, whereas the attention in the remaining four settings is more scattered. This feature makes INJECTION our primary candidate for self-assessing 3D object detection performance with real-world data.

3.7.2 Self-Assessment of Lead Vehicle Distance Estimation:

Accurate distance estimation to the lead vehicle is essential for a range of automated driving functions, such as ACC, collision warning systems, and automated emergency braking. Various methods for distance estimation have been extensively explored in the literature. Monocular cameras, as discussed by [145], are widely used due to their

simplicity and cost-effectiveness, but they are particularly vulnerable in low-light conditions. Stereo cameras [146], which provide depth information by analysing the disparity between two camera images, offer improved distance estimation over monocular systems but also struggle in poor lighting and are computationally more demanding. Radar sensors [147] are commonly adopted, particularly in highway ACC, due to their robustness under adverse weather. However, they often suffer from limitations in urban environments where their relatively low spatial resolution and susceptibility to noise and signal clutter can hinder accurate distance measurements. LiDAR-based systems provide a more precise alternative for distance estimation with their higher range accuracy and the ability to generate detailed 3D representations in complex driving scenarios. LiDARs can effectively capture both the shape and position of surrounding vehicles and obstacles, making it a preferred option in urban driving where accuracy and comprehensive environmental perception are critical.

In the following, a deep learning-based SA system is developed to evaluate the predictions of a LiDAR-based lead vehicle distance estimation system. The SA model takes as input the activation maps of the main model and predicts a trust indicator for the main model's predictions.

Framework description

Figure 57 illustrates the block diagram of the SA framework. The distance estimation system includes the CenterPoint 3D object detector [148] and a lead vehicle filter. The filter first identifies the closest bounding box to the Ego vehicle on its planned waypoints, designating it as the lead vehicle. It then calculates the distance of the lead vehicle bounding box's center along the waypoints' path. If no lead vehicle is detected, the estimated distance is set to a maximum threshold. As illustrated in Figure 57, the SA model takes as input the early layer activation map generated by the point cloud processor. This activation map contains embedded information from the point cloud input, reflecting how the object detector interprets it. The ResNet18 model is used to extract relevant features from this input data. Finally, a Fully Connected Layer generates a binary output that classifies whether the distance estimator's output is trustworthy or not. It should be noted that the SA model for trustworthy distance estimation illustrated in Figure 57 is different from that illustrated in Figure 1 for SA of 3D object detection, although the two models have similar architectures.

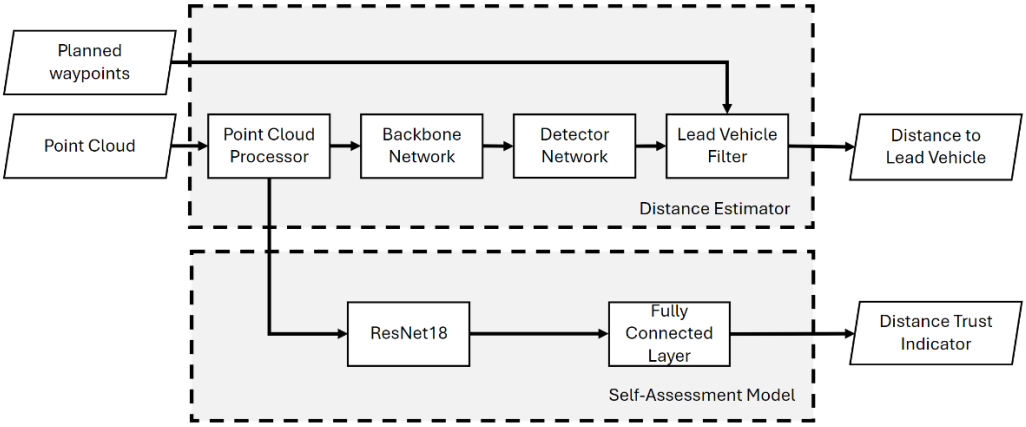


Figure 57: Distance estimation to the lead vehicle and SA framework generating a binary distance trust indicator during inference.

The SA model for the distance estimation is trained using trust labels generated for each input frame. These labels are determined by comparing the ground truth distance to the lead vehicle with the estimated distance provided by the estimator. A distance error threshold is defined to specify the acceptable level of error for the subject application, e.g., urban chauffeur.

Qualitative evaluation example

Figure 58 shows an example of input point cloud data, the lead vehicle filter, and the estimated SA label at the top-left corner. The lead vehicle filter, indicated by the dashed lines, covers areas along the planned trajectory, extending 50 meters longitudinally and 1.5 meters laterally. Note that spatial filtering is not applied to the point cloud in this case. The solid black line is the planned trajectory of the ego-vehicle, and the green and red bounding boxes correspond to the ground-truth and predicted bounding boxes of the lead vehicle, respectively. A distance error threshold of 0.1 meters has been selected to meet the high-accuracy localisation requirements for urban driving. The sample illustrated in Figure 58 is labelled “not trusted” by the SA framework, as the distance estimation error is more than 0.1 meters in this case.

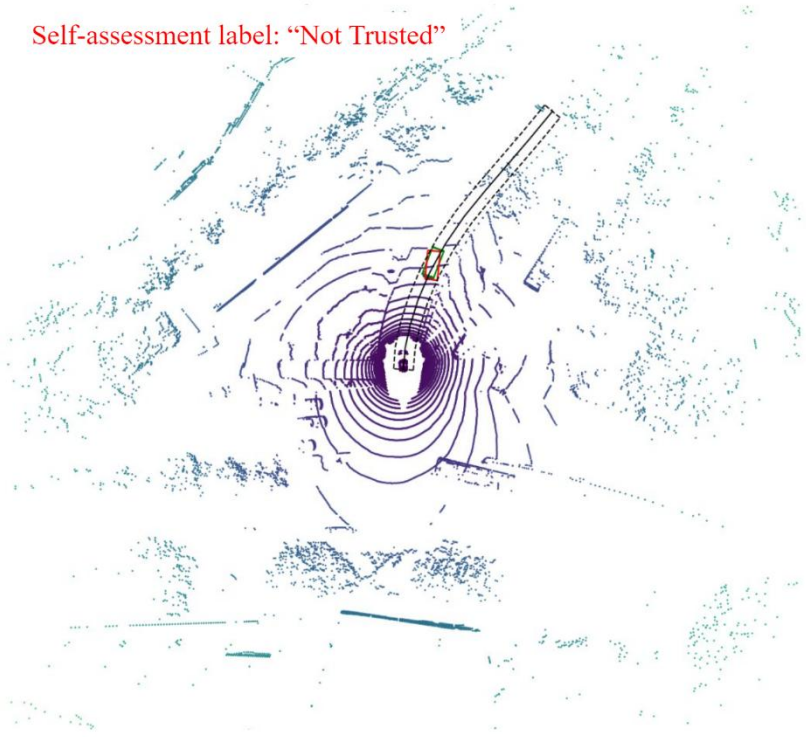


Figure 58: An example of a point cloud sample and lead vehicle filter indicated by dashed lines.

3.8 EXP8 (PERCIV): Emergency evasion manoeuvre under adverse weather conditions including perception self-assessment

3.8.1 Introduction

EXP 8 (“Emergency evasion manoeuvre under adverse weather conditions including perception self-assessment”) has the objective is to perform a collision avoidance manoeuvre (e.g. with leading vehicle, cyclist, etc.) in poor weather conditions on a potentially slippery road surface.

Rain can influence the ego-vehicle in multiple ways. First, it makes perception harder as rain droplets and water stirred up by the tires hamper most automotive sensors' performance. Second, a wet, slippery surface could lead to vehicle instability, especially at the edge of its friction limits.

Thus, it is not surprising that rain and wet pavement together cause significantly more accidents/fatalities (US average: 1400k/6k, based on U.S. Department of Transportation, 2007-2016) than snowy, icy, and foggy conditions together (570k/2.3k).

With this motivation, in this experiment, Perciv AI and TUD will in later WPs present a "full stack" solution, i.e., starting from perception and understanding the environment (Perciv AI), to vehicle control (TUD) in rainy / wet pavement conditions. See Figure 59 for an overview of the scenario.

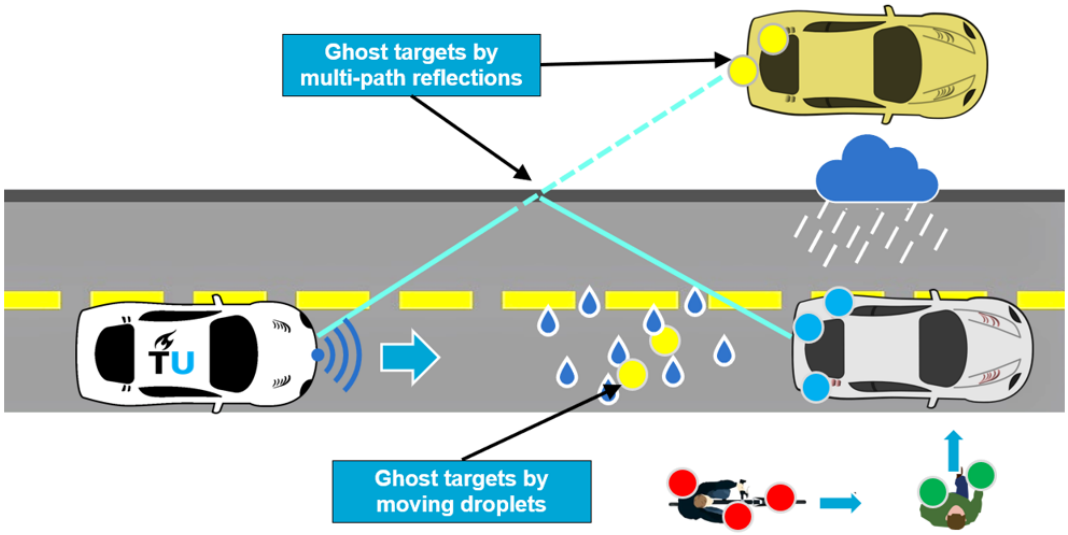


Figure 59: High level overview of EXP 8's scenario and main challenges.

The high-level steps of this experiment are:

1. TUD and PercivAI will create rainy conditions on a test track/dedicated road to replicate the desired scenario in a safe way.
2. PercivAI will collect multimodal datasets similar to [149], including next generation 4D radar sensors, cameras, LiDARs, and GNSS systems using the artificially created and real rainy scenarios to train and tune their perception algorithms.
3. PercivAI will use the collected data to develop novel, AI-driven radar perception algorithms, which will filter the input in multiple ways (e.g. ghost vs real radar points) and output a list of objects and estimated ego-motion/odometry information based on the weather robust radar sensors. These outputs will be an input to TUD's motion control module.
4. Integrate a full stack pipeline into a TUD research vehicle, including the perception, the motion control, and their communication. See Figure 60 for an overview of the integrated pipeline.

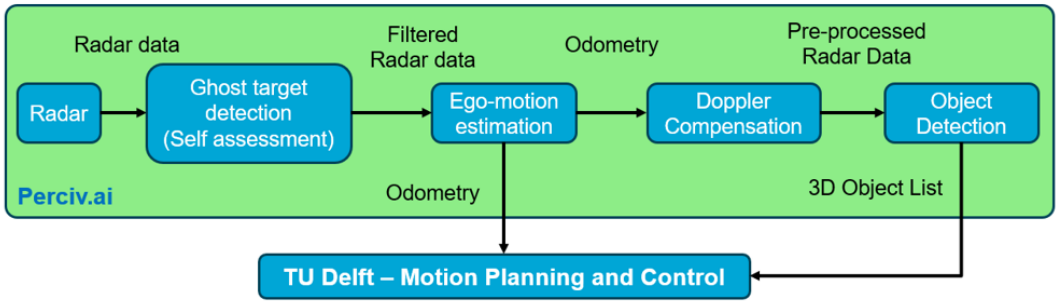


Figure 60: Architecture of EXP8 with corresponding contributions of PercivAI and TUD.

Given the scope of this document (D.3.2: Perception System and Self-Assessment), here we will focus on the description of the perception modules of the experiment, i.e., PercivAI's work. In Work Package WP3: "Perception", PercivAI participates in Task T3.2: "Semantic Scene Analysis and Precise Localization", targeting scenarios with heavy raining and wet surfaces, responsible for two main components: Scene segmentation and Localization.

3.8.2 Scene Segmentation

Segmentation of ghost targets, noise suppression

In the scenario targeted in EXP 8, droplets are in the air not only from the rain itself, but also from leading vehicles which stir up the water from the road surfaces. Both kinds of droplets can disadvantageously influence all types of sensors available for intelligent vehicles: cameras, LiDARs, and even radars. Furthermore, radars are known to be noisy and sensitive to multipath effects, reporting reflections at incorrect locations, see Figure 59. Thus, the perception module needs to filter or, in other words, segment ghost and real reflections to suppress noise.

Segmentation of moving targets

After segmenting real and ghost target, it is important to segment which parts of the scene are moving, and which parts are static. This distinction is highly beneficial for many use cases, e.g., detecting and predicting future positions of moving objects nearby or using the static environment to reliably estimate the ego-motion of the vehicle. Furthermore, in general, explicitly paying more attention to moving objects can literally save lives, as these objects are usually road users in the addressed scenarios. One important aspect to discuss is that radar can measure velocity of its reported reflections, which may suggest that this task is straightforward. However, radar reports only *relative* and only *radial* velocities, while we are interested in moving objects in the absolute sense, even if they move tangentially, similarly to [150] or [151].

Segmentation by class

After segmenting real, moving targets, we proceed to classify them semantically. That is, we must assign a road user class (e.g. car, cyclist, pedestrian) to the segmented moving radar points. This is beneficial to better estimate their future movements and plan around them in a safe and reliable way as explained in [152], [153].

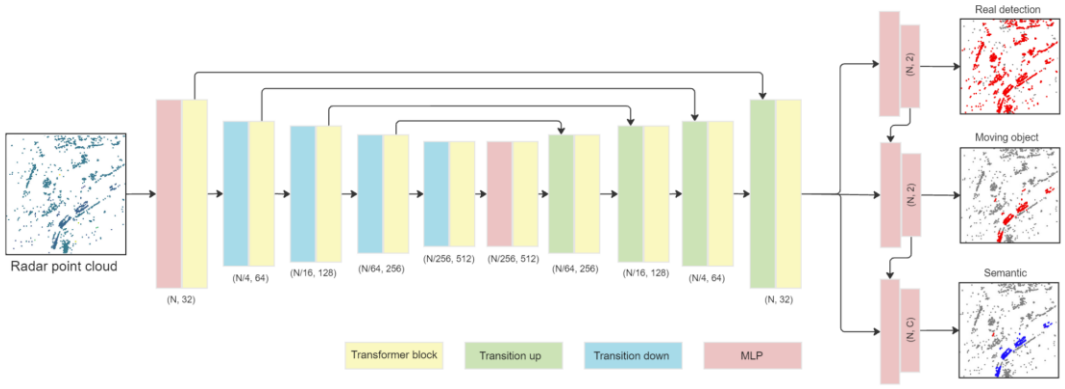


Figure 61: Proposed architecture of the multipurpose, sequential radar segmentation network.

Proposed Solution

Realizing that the subtasks discussed above are sequential in nature, PercivAI designed, developed, and implemented a radar point cloud segmentation network which exploits this, see Figure 61. The algorithm takes a radar point cloud as an input, then segments *real targets,* of those, the *moving targets,* and finally, of those, the *classes of interests.* The novelty of the method lies in the fact that unlike the SOTA methods ([150], [151], [152], [153]), it *performs all three tasks in a single network.* See Figure 62 for qualitative results from the multitask network, which shows example outputs of our network performing real target, moving target, and semantic segmentation at the same time. Note how the network can detect the pedestrian and the car in the shadows, which would be challenging for a camera.

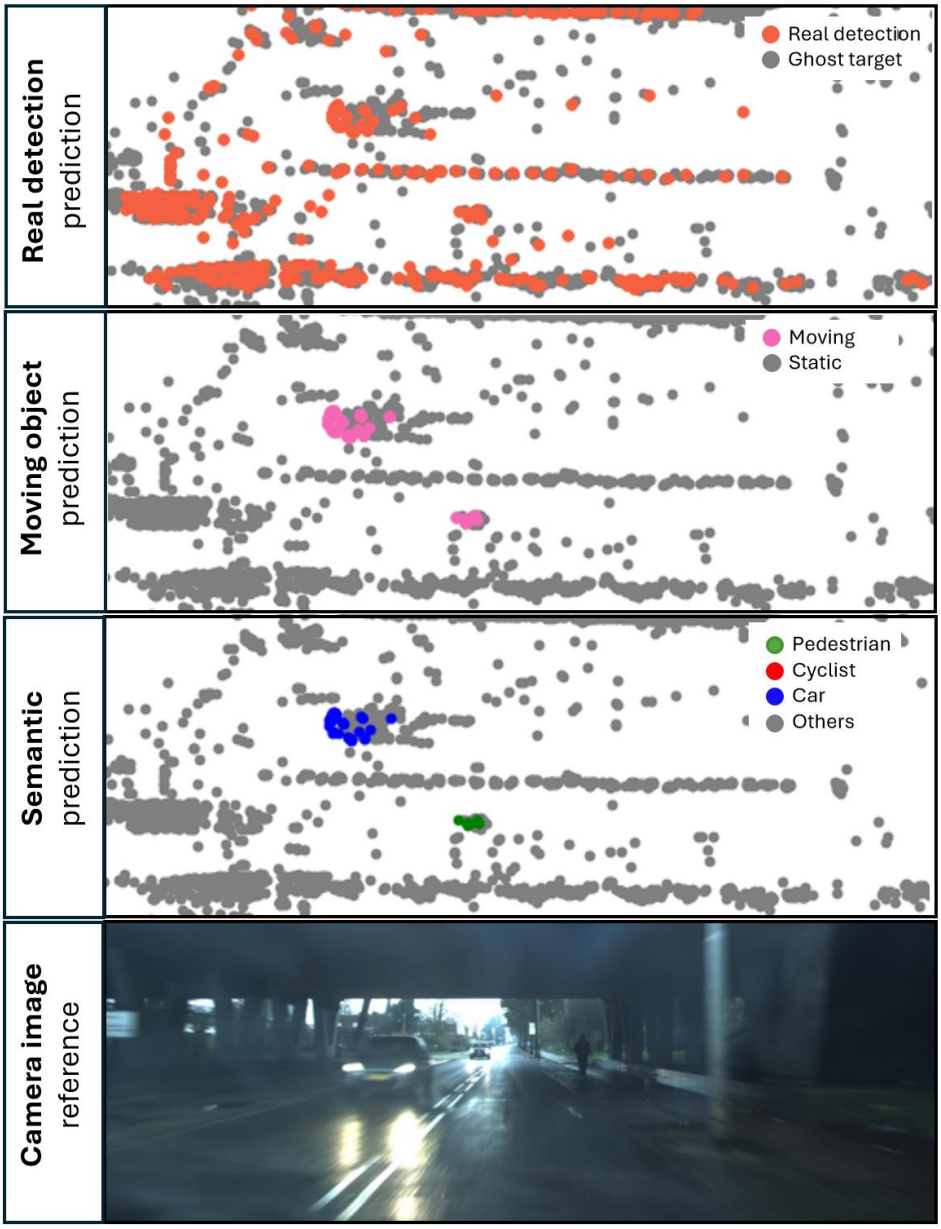


Figure 62: Qualitative results from the multitask network

3.8.3 Localization

As shown in Figure 60, PercivAI is expected to provide not just a scene understanding, but also an estimation of the ego-vehicle's odometry. As explained in Section 0, segmenting moving, and static points is highly beneficial for this tasks, as static points can be used as "anchor points" to which we can compare and measure the ego-vehicle's motion. It is worth mentioning that by utilizing the output of the network introduced above we also save computational resources.

The sequence of static radar point clouds is then fed to a point cloud-based SLAM method, KISS-ICP [154], modified for radar data in terms of input channels, expected density, and trust in potentially multipath reflections.

As a result, our pipeline can accurately estimate the odometry of the ego-vehicle using purely the radar point cloud as an input; see Figure 63 and Figure 64 below for qualitative examples. Top-left shows the camera image, while the other two views show the current radar scan (dark green points) and the accumulated, mapped points, nicely forming the map of TUD campus (green-yellow, colored by time).

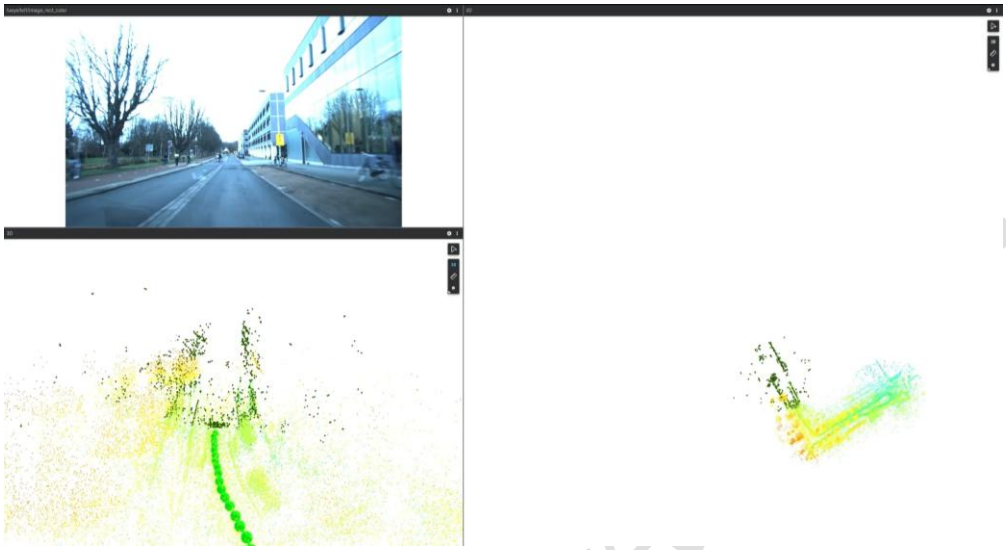


Figure 63: Qualitative result of radar based ego-motion estimation, Example I.

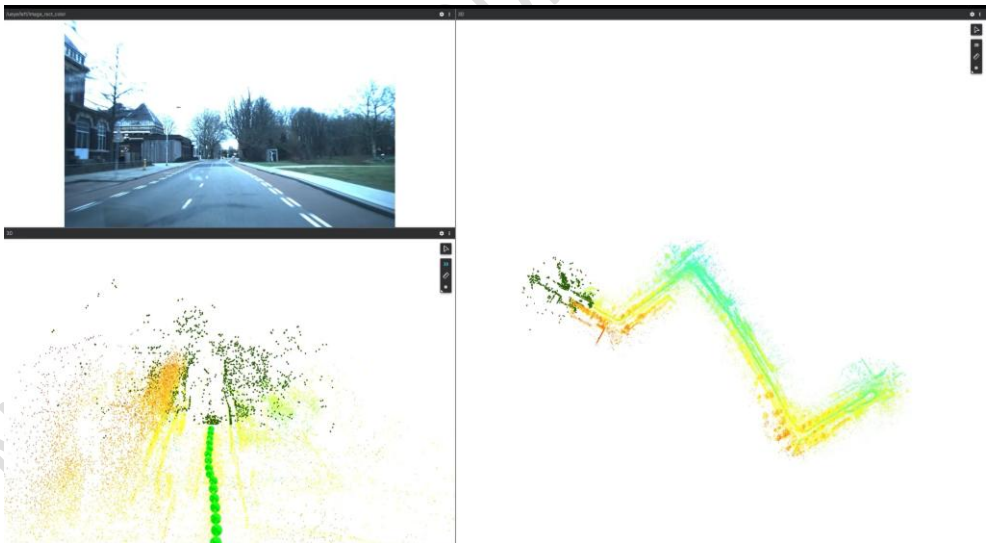


Figure 64: Qualitative result of radar based ego-motion estimation, Example II.

4. Conclusions

This Deliverable 3.2 concludes the report on the work performed within WP3 in the EVENTS project, which was partially described in D3.1 [25]. The aim of this work package is to provide the machine perception system and SA capabilities needed to

facilitate the various experiments (EXP1-EXP8), as these are specified by the EVENTS project partners in D2.1 [82] and D2.2 [83].

In **T3.1**, we explored the use of existing public datasets, described a newly acquired road debris dataset, covered data generation based on manual annotation and simulation, and presented an approach for self-supervised learning for object detection.

In **EXP1** a LiDAR-based environment perception pipeline was developed, where a deep learning-based detector detects objects from a set of predefined classes, e.g., car, bike, and pedestrian. We also use a traditional clustering approach to segment generic objects that are not part of the above set of classes. An object merger combines the detections of both methods and discards near-identical duplicate detections. A multi-object tracker tracks detected objects across time, and a motion prediction component predicts the future path of each tracked object. As part of WP5, we aim to integrate the more sophisticated map-based PGP motion-prediction of [156], which allows multi-modality in the prediction.

In **EXP2**, a cooperative motion prediction framework was developed. The core prediction model is based on the SOTA model HiVT [13] without relying on a specified map for general applicability. Two distinct methods (Euclidean and BBox clustering) were evaluated for the association of multiple overlapping detections. The results demonstrate that V2V-enhanced predictions achieve a better understanding of the traffic scene.

To support V2V information exchange, TECN and ICCS jointly designed custom JSON file formats that host the information disseminated by (custom) CAMs and CPMs. The specification of the corresponding JSON structure and data field definitions were based on ETSI documents [16-19].

A novel algorithm for probabilistic fusion of CAM and CPM information was designed, aiming at end-to-end explainability, parameter interpretability and the provision of inherent reliability indicators for the output. The algorithm uses non-linear particle localization filters and employs a custom ray-tracing algorithm for FoV calculation, which facilitates information consistency checks. The derived output is a probabilistic occupancy grid, inherently accounting for the locality of (collective) perception reliability.

TECN will continue the development of the cooperative motion prediction framework towards additional consideration of detector confidence and extending the study in other collaborative domains within simulated environments. ICCS will further adjust and test the proposed CAM/CPM fusion algorithm in custom CARLA scenarios, with camera/LiDAR fusion as an elementary perception stack. Setting the derived

occupancy probabilities as priors for the next time step will be also studied as tracking scheme during the module evaluation work in WP6. TECN and ICCS will continue joint efforts to integrate the probabilistic occupancy grids generated by ICCS into the V2X communication system in a hybrid setup developed by TECN. These occupancy grids, communicated via the ROS nav_msgs/OccupancyGrid format, will be derived from the 3D detections made by the individual perception stacks of the involved CAVs and will provide a detailed map of the environment. This will enhance the situational awareness of the CAVs, providing a more accurate representation of the surrounding space and potential obstacles.

In **EXP3**, a SA framework has been developed and evaluated for object-tracking algorithms based on subjective logic, tested both in simulation and on real-world data. These methods were presented at scientific conferences, including work by Griebel et al.[87][88]. Future efforts as part of WP5 will focus on integrating these SA approaches into UULM's test vehicle for real-world applications to enhance safety and robustness in autonomous driving.

For **EXP4**, we have developed a pipeline for updating pre-existing HD-map under road work conditions. Our model assumes traffic bollards are being used to separate drivable vs non-drivable lanes, where such bollards are then used to determine the updated lane boundary. We will provide quantitative results regarding the accuracy of the updated HD-map in WP6.

In **EXP5**, HIT and TECN in collaboration designed and implemented a predictive perception system that can reliably detect and track multiple 3D objects moving at various speeds in real-time (Hitachi's contribution), and forecast their future movements based on historical trajectories (TECN's contribution). WMG also contributed to this experiment a SA mechanism for the perception system. This combined system provides the ego-vehicle with quick and reliable information for safe decision-making at intersections. These modules were developed and tested using data collected by Hitachi's demo vehicle. In WP6, we will perform a more quantitative evaluation of the developed algorithms and aim to integrate the developed algorithm into the demo vehicle in WP5.

In **EXP6,** the data collection of a new debris dataset was accomplished. An overdriveability classifier has been trained on the gathered data. Additionally, the performance of the perception stack was qualitatively described with examples. Our next steps are to perform a quantitative analysis of the perception algorithm performance in WP6 and to integrate it on the demo vehicle in WP5.

Within **EXP7**, SA mechanisms for LiDAR-based 3D object detection and relative localization to the leading vehicle were developed. These mechanisms were evaluated using public datasets and demonstrated superior performance compared to the

current state-of-the-art. The next step involves implementing these mechanisms into WMG's experimental vehicle platform (WP5) and evaluating their performance using real-world data (WP6).

Within **EXP8**, methods were developed for scene segmentation (covering noise, movement, and semantics) and ego-motion estimation based solely on radar, making the solution highly robust to weather conditions. The aim is integration in WP5 with TUD's work on control in WP4, and quantitative testing in WP6.

References

[1] https://github.com/NVlabs/imaginaire/blob/master/MODELZOO.md#unsupervised-image-to-image-translation

[2] https://github.com/NVlabs/imaginaire/blob/master/projects/unit/README.md

[3] Jeong-gi Kwak, Youngsaeng Jin, Yuanming Li, Dongsik Yoon, Donghyeon Kim, Hanseok Ko. Adverse Weather Image Translation with Asymmetric and Uncertainty-aware GAN. https://arxiv.org/abs/2112.04283, github : https://github.com/jgkwak95/AU-GAN

[4] V. Mușat, I. Fursa, P. Newman, F. Cuzzolin and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 2906-2915, doi: 10.1109/ICCVW54120.2021.00325.

[5] Tim Brooks, Aleksander Holynski, Alexei A. Efros. "InstructPix2Pix: Learning to Follow Image Editing Instructions." https://arxiv.org/abs/2211.09800

[6] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, Stefano Ermon. "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations." https://arxiv.org/abs/2108.01073

[7] H. Caesar et al., «nuscenes: A multimodal dataset for autonomous driving», en Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621-11631

[8] "ASAM OpenSCENARIO XML 1.3.0 Specification". ASAM. https://publications.pages.asam.net/standards/ASAM_OpenSCENARIO/ASAM_OpenSCENARIO_XML/latest/index.html.

[9] "ASAM OpenDRIVE 1.8.0 Specification". ASAM. https://publications.pages.asam.net/standards/ASAM_OpenDRIVE/ASAM_OpenDRIVE_Specification/latest/specification/index.html.

[10] https://carla.readthedocs.io/en/latest/tuto_G_scenic/

[11] Mapillary Dataset: https://www.mapillary.com/

[12] Zenseact Dataset: https://zod.zenseact.com/

[13] D. Dwibedi, I. Misra and M. Hebert, "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection," IEEE International Conference on Computer Vision, pp. 1310-1319, 2017.

[14] G. Ghiasi et al., "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2917-2927, 2021.

[15] H. Caesar et al., «nuscenes: A multimodal dataset for autonomous driving», en Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621-11631.

[16] P. Sun et al., «Scalability in Perception for Autonomous Driving: Waymo Open Dataset», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), jun. 2020.

[17] M.-F. Chang et al., «Argoverse: 3d tracking and forecasting with rich maps», en Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 8748-8757.

[18] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, y J. Ma, «Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication», en 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 2583-2589.

[19] Y. Li et al., «V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving», IEEE Robot. Autom. Lett., vol. 7, n.o 4, pp. 10914-10921, 2022.
[20] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, y J. Ma, «V2x-vit: Vehicle-to-everything cooperative perception with vision transformer», en European conference on computer vision, Springer, 2022, pp. 107-124.
[21] W. Zimmer, C. Creß, H. T. Nguyen, y A. C. Knoll, «A9 Intersection Dataset: All You Need for Urban 3D Camera-LiDAR Roadside Perception», ArXiv Prepr. ArXiv230609266, 2023.
[22] H. Yu et al., «Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 21361-21370.
[23] H. Yu et al., «V2X-Seq: A Large-Scale Sequential Dataset for Vehicle-Infrastructure Cooperative Perception and Forecasting», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5486-5495.
[24] R. Xu et al., «V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13712-13722.
[25] D. Gavrila et al., "EVENTS D.3.1 Perception Components Methods," December 2023. [Online]. Available: https://www.events-project.eu/wp-content/uploads/2024/10/D3.1.pdf. [Accessed 22 October 2024].
[26] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In Computer Vision and Pattern Recognition (CVPR), pages 17853–17862, 2023.
[27] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S.C.H. Hoi. Deep learning for person re-identification: A survey and outlook. Transactions on Pattern Analysis and Machine Intelligence (PAMI), 44(6):2872–2893, 2021.
[28] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. RT- 2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning (CoRL), pages 2165–2183, 2023.
[29] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In International Conference on Computer Vision (ICCV), pages 9650–9660, 2021.
[30] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (TMLR), pages 2835–8856, 2024.
[31] O. Siméoni, G. Puy, H.V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet, and J. Ponce. Localizing objects with self-supervised transformers and no labels. In British Machine Vision Conference (BMVC), 2021.
[32] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J.M. Alvarez. FreeSOLO: Learning to segment objects without annotations. In Computer Vision and Pattern Recognition (CVPR), pages 14176–14186, 2022.
[33] Y. Wang, Y. Chen, and Z. Zhang. 4D unsupervised object discovery. In Neural Information Processing Systems (NeurIPS), pages 35563–35575, 2022.
[34] Y. You, K. Luo, C.P. Phoo, W. Chao, W. Sun, B. Hariharan, M. Campbell, and K.Q. Weinberger. Learning to detect mobile objects from LiDAR scans without labels. In Computer Vision and Pattern Recognition (CVPR), pages 1130–1140, 2022.

[35] Lentsch, T., Caesar, H., & Gavrila, D. M. (2024). UNION: Unsupervised 3D Object Detection using Object Appearance-based Pseudo-Classes. In Neural Information Processing Systems (NeurIPS), 2024.
[36] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. In Computer Vision and Pattern Recognition (CVPR), pages 11621–11631, 2020.
[37] L. McInnes, J. Healy, and S. Astels. HDBSCAN: Hierarchical density based clustering. Journal of Open Source Software (JOSS), 2(11):205, 2017.
[38] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In Computer Vision and Pattern Recognition (CVPR), pages 11784–11793, 2021.
[39] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindstr¨om, Daria Mo- torniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In International Conference on Computer Vision (ICCV), pages 20178–20188, 2023.
[40] Zhihao Shen, Huawei Liang, Linglong Lin, Zhiling Wang, Weixin Huang, and Jie Yu. Fast ground seg- mentation for 3D LiDAR point cloud based on jump- convolution-process. Remote Sensing, 13(16):3239, 2021.
[41] Radu Bogdan Rusu. Semantic 3D object maps for everyday manipulation in human living environments. KI-K¨unstliche Intelligenz, 24:345–348, 2010.
[42] Xiao Zhang, Wenda Xu, Chiyu Dong, and John M. Dolan. Efficient L-shape fitting for vehicle detection using laser scanners. In Intelligent Vehicles (IV), pages 54–59, 2017.
[43] James B Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. Mathematical Programming, 78:109–129, 1997.
[44] Congchao Wang, Yizhi Wang, Yinxue Wang, Chiung- Ting Wu, and Guoqiang Yu. muSSP: Efficient min- cost flow algorithm for multi-object tracking. In Neural Information Processing Systems (NeurIPS), volume 32, 2019.
[45] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehi- cles with embedded systems. In International Conference on Cyber-Physical Systems (ICCPS), pages 287–296, 2018.
[46] H. Caesar et al., «nuscenes: A multimodal dataset for autonomous driving», en Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621-11631.
[47] P. Sun et al., «Scalability in Perception for Autonomous Driving: Waymo Open Dataset», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), jun. 2020.
[48] M.-F. Chang et al., «Argoverse: 3d tracking and forecasting with rich maps», en Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 8748-8757.
[49] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, y J. Ma, «Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication», en 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 2583-2589.
[50] Y. Li et al., «V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving», IEEE Robot. Autom. Lett., vol. 7, n.o 4, pp. 10914-10921, 2022.
[51] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, y J. Ma, «V2x-vit: Vehicle-to-everything cooperative perception with vision transformer», en European conference on computer vision, Springer, 2022, pp. 107-124.

[52] W. Zimmer, C. Creß, H. T. Nguyen, y A. C. Knoll, «A9 Intersection Dataset: All You Need for Urban 3D Camera-LiDAR Roadside Perception», ArXiv Prepr. ArXiv230609266, 2023.
[53] H. Yu et al., «Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 21361-21370.
[54] H. Yu et al., «V2X-Seq: A Large-Scale Sequential Dataset for Vehicle-Infrastructure Cooperative Perception and Forecasting», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5486-5495.
[55] R. Xu et al., «V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13712-13722.
[56] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, y V. Koltun, «CARLA: An open urban driving simulator», en Conference on robot learning, PMLR, 2017, pp. 1-16.
[57] J. Araluce, A. Justo, A. Arizala, L. González, y S. Díaz, «Enhancing Motion Prediction by a Cooperative Framework», en 2024 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2024, pp. 1389-1394.
[58] Z. Zhou, L. Ye, J. Wang, K. Wu, y K. Lu, «Hivt: Hierarchical vector transformer for multi-agent motion prediction», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8823-8833.
[59] J. Schmidt, J. Jordan, F. Gritschneder, y K. Dietmayer, «Crat-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention», en 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 7799-7805.
[60] A. Cui, S. Casas, K. Wong, S. Suo, y R. Urtasun, «Gorela: Go relative for viewpoint-invariant motion forecasting», en 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 7801-7807.
[61] ETSI TS 302 637-2 (on CAM).
[62] ETSI TS 103 324 (on CPM).
[63] ETSI TR 103 562 (on CPM).
[64] ETSI TS 102 894-2 on the Common Data Dictionary used in CAM/CPM.
[65] M. Shan, S. Worrall E. Nebot. Development and Demonstrations of Cooperative Perception for Connected and Automated Vehicles. Report, iMOVE CRC Project, 2021.
[66] Godoy, J.; Jiménez, V.; Artuñedo, A.; Villagra, J. A Grid-Based Framework for Collective Perception in Autonomous Vehicles. Sensors 2021, 21, 744.
[67] Nadia Mouawad, Valérian Mannoni. Collective perception messages: new low complexity fusion and V2X connectivity analysis. pp.1-5, 2021, 2021 IEEE 94th Vehicular Technology Conference.
[68] Crolla, David, David Foster, et al. Encyclopedia of Automotive Engineering, Volume 4. John Wiley & Sons Ltd, 2015.
[69] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.
[70] O. Cappe, S. J. Godsill and E. Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," in Proceedings of the IEEE, vol. 95, no. 5, pp. 899-924, 2007.
[71] M. Sun, M. Li and R. Gerdes, "A data trust framework for VANETs enabling false data detection and secure vehicle tracking," 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 2017, pp. 1-9.

[72] J. Zhang, I. B. Jemaa and F. Nashashibi, "Trust Management Framework for Misbehavior Detection in Collective Perception Services," 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Singapore, 2022, pp. 596-603, doi: 10.1109/ICARCV57592.2022.10004259.

[73] R. W. van der Heijden, S. Dietzel, T. Leinmüller and F. Kargl, "Survey on Misbehavior Detection in Cooperative Intelligent Transportation Systems," in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 779-811, Firstquarter 2019, doi: 10.1109/COMST.2018.2873088.

[74] S. So, P. Sharma and J. Petit, "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 564-571, doi: 10.1109/ICMLA.2018.00091.

[75] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa and P. Urien, "Simulation Framework for Misbehavior Detection in Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 6631-6643, June 2020, doi: 10.1109/TVT.2020.2984878.

[76] Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons.

[77] Benmahammed, I. (2022, April). How to add confidence to your Machine Learning models. (TotalEnergies Digital Factory) Retrieved October 2023, from https://medium.com/totalenergies-digital-factory/how-to-add-confidence-to-your-machine-learning-models-b1228217858e

[78] Buerkle, C., Oboril, F., Jarquin, J., & Scholl, K. (2020). Efficient dynamic occupancy grid mapping using non-uniform cell representation. IEEE IV 2020, Proceedings 1629–1634.

[79] Cloud, S. (2023, June). How to Get a Confidence Measure for Each Prediction in a Machine Learning Model Python. Retrieved October 2023, from https://saturncloud.io/blog/how-to-get-a-confidence-measure-for-each-prediction-in-a-machine-learning-model-python/

[80] Dongkai, W., & Zhang, S. (2022). Contextual instance decoupling for robust multi-person pose estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[81] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & V. Koltun. (2017). CARLA: An open urban driving simulator. Proceedings of the 1st Annual Conference on Robot Learning.

[82] EVENTS. (2023). Deliverable D2.1: User and system requirements for selected use cases.

[83] EVENTS. (2023). Deliverable D2.2: Full Stack Architecture & Interfaces.

[84] Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. 2012 IEEE conference on computer vision and pattern recognition.

[85] Griebel, T., & al., e. (2022). Self-Assessment for Single-Object Tracking in Clutter Using Subjective Logic. 2022 25th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION49751.2022.9841294. Linköping, Sweden.

[86] Griebel, T., & al., e. (2022). Self-Assessment for Single-Object Tracking in Clutter Using Subjective Logic. 2022 25th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION49751.2022.9841294. Linköping, Sweden.

[87] Griebel, T., Dehler, N., Scheible, A., Buchholz, M., & Dietmayer, K. (2024). Self-Assessment for Multi-Object Tracking Based on Subjective Logic. 2024 IEEE Intelligent Vehicles Symposium (IV). Jeju Island, Korea, Republic of.

[88] Griebel, T., Heinzler, J., Buchholz, M., & Dietmayer, K. (2023). Online Performance Assessment of Multi-Sensor Kalman Filters Based on Subjective Logic. 2023 26th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION52260.2023.10224188. Charleston, SC, USA.

[89] Griebel, T., Heinzler, J., Buchholz, M., & Dietmayer, K. (2023). Online Performance Assessment of Multi-Sensor Kalman Filters Based on Subjective Logic. 2023 26th International Conference on Information Fusion (FUSION). Charleston, SC, USA.

[90] Griebel, T., Müller, J., Buchholz, M., & Dietmayer, K. (2020). Kalman Filter Meets Subjective Logic: A Self-Assessing Kalman Filter Using Subjective Logic. 2020 IEEE 23rd International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION45008.2020.9190520. Rustenburg, South Africa.

[91] Müller, J., Griebel, T., Gabb, M., and Buchholz, M. (2019). Subjective Logic-Based Identification of Markov Chains and Its Application to CAV's Safety. 2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS), Honolulu, HI, USA, 2019, pp. 1-5, doi: 10.1109/CAVS.2019.8887769.

[92] Wodtko, T., Griebel, T., and Buchholz, M. (2023). Adaptive Patched Grid Mapping. 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023, pp. 306-313, doi: 10.1109/ITSC57777.2023.10422649.

[93] Scheible, A., Griebel, T., Herrmann, M., Hermann, C., and Buchholz, M. (2023). Track Classification for Random Finite Set Based Multi-Sensor Multi-Object Tracking. 2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI), Bonn, Germany, 2023, pp. 1-8, doi: 10.1109/SDF-MFI59545.2023.10361438.

[94] Hermann, C., Herrmann, M., Griebel, T., Buchholz, M., & Dietmayer, K. (2023). The Fast Product Multi-Sensor Labeled Multi-Bernoulli Filter. 2023 26th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION52260.2023.10224189. Charleston, SC, USA.

[95] Wodtko, T., Griebel, T., Scheible, A., and Buchholz, M. (2024). Conflict Handling in Time-Dependent Subjective Networks. 2024 27th International Conference on Information Fusion (FUSION), Venice, Italy, 2024, pp. 1-8, doi: 10.23919/FUSION59988.2024.10706464.

[96] Griebel, T., Müller, J., Buchholz, M., and Dietmayer, K. (2024). Adaptive Kalman Filtering Based on Subjective Logic Self-Assessment. 2024 27th International Conference on Information Fusion (FUSION), Venice, Italy, 2024, pp. 1-8, doi: 10.23919/FUSION59988.2024.10706328.

[97] Scheible, A., Griebel, T., and Buchholz, M. (2024). Self-Monitored Clutter Rate Estimation for the Labeled Multi-Bernoulli Filter. 2024 27th International Conference on Information Fusion (FUSION), Venice, Italy, 2024, pp. 1-7, doi: 10.23919/FUSION59988.2024.10706463.

[98] Holger, C., & al., e. (2020). nuscenes: A multimodal dataset for autonomous driving. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

[99] Holzbock, A., Tsaregorodtsev, A., & Belagiannis, V. (2023). Pedestrian Environment Model for Automated Driving. arXiv preprint arXiv:2308.09080.

[100] (2019). ISO/PAS 21448: Road vehicles - Safety of the intended functionality. International Organization for Standardization.

[101] Jøsang, A. (2016). Subjective Logic: A formalism for reasoning under uncertainty. Springer Publishing Company, Incorporated.

[102] Lecture 15: Gaussian Processes. . (n.d.). (Cornell Bowers CIS) Retrieved October 12, 2023, from https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html
[103] Nuss, D., Reuter, S., Thom, M., Yuan, T., Krehl, G., Maile, M., . . . Dietmayer, K. (2018). A random finite set approach for dynamic occupancy grid maps with real-time application. The International Journal of Robotics Research, vol. 37, no. 8, pp. 841–866.
[104] Ohazulike et al., A. (2023, June). EVENTS D.2.2 Full Stack Architecture & Interfaces. Retrieved October 13, 2023, from https://www.events-project.eu/wp-content/uploads/2023/07/EVENTS_D2.2_Full-Stack-Architecture-Interfaces_v1.0.pdf
[105] P. Cong, X. Z. (2022). STCrowd: A multimodal dataset for pedestrian perception in crowded scenes. Computer Vision and Pattern Recognition (CVPR) (pp. 19608–19617). IEEE/CVF.
[106] Paek, D.-H., Kong, S.-H., & Wijaya, K. T. (2022). K-Radar: 4D Radar Object Detection for Autonomous Driving in Various Weather Conditions. Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track.
[107] Scheiner, N., Kraus, F., Appenrodt, N., Dickmann, J., & Sick, B. (2021). Object detection for automotive radar point clouds--a comparison. AI Perspectives, 3(1), 1--23.
[108] Schumann, O., Wöhler, C., Hahn, M., & Dickmann, J. (2017). "Comparison of random forest and long short-term memory network performances in classification tasks using radar,". 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF).
[109] Tango et al., F. (2023, March 2023). D.2.1: User and System Requirements for selected Use-cases. Retrieved October 26, 2023, from https://www.events-project.eu/wp-content/uploads/2023/06/EVENTS_D2.1_User-and-System-Requirements-for-selected-Use-cases_v1.0_with-requirements.pdf
[110] Tutorial: Basic Statistics in Python — Probability. (2018, July 18). (DATAQUEST) Retrieved October 13, 2023, from https://www.dataquest.io/blog/basic-statistics-in-python-probability/
[111] Tyagi, K., Zhang, S., Zhang, Y., Kirkwood, J., Song, S., & Manukian, N. (2023). Machine Learning Based Early Debris Detection Using Automotive Low Level Radar Data. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
[112] Wellhausen, C., Clemens, J., & Schill, K. (2021). Efficient grid map data structures for autonomous driving in large-scale environments. IEEE ITSC 2021, Proceedings, 2855–2862.
[113] Wodtko, T., Griebel, T., & Buchholz, M. (2023). Adaptive Patched Grid Mapping. arXiv preprint arXiv:2308.03416.
[114] How we make our HD Maps | TomTom Newsroom
[115] Q. Li, Y. Wang, Y. Wang, H. Zhao, ‘‘HDMapNet: An Online HD Map Construction and Evaluation Framework’’, IEEE International Conference on Robotics and Automation, 2022.
[116] Y. Liu, T. Yuan, Y. Wang, Y. Wang, H. Zhao, ‘‘VectorMapNet: End-to-end Vectorized HD Map Learning’’, International conference on machine learning, 2023.
[117] Guo Y, Zhou J, Li X, Tang Y, Lv Z. A Review of Crowdsourcing Update Methods for High-Definition Maps. ISPRS International Journal of Geo-Information. 2024
[118] Z. Zhou, L. Ye, J. Wang, K. Wu, y K. Lu, «Hivt: Hierarchical vector transformer for multi-agent motion prediction», en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8823-8833.

[119] S. Shi, X. Wang, and H. Li, “Pointrcnn: 3d object proposal generation and detection from point cloud,” in CVPR, 2019, pp. 770–779.
[120] J. Liu, Y. Chen, X. Ye, Z. Tian, X. Tan, and X. Qi, “Spatial pruned sparse convolution for efficient 3d object detection,” arXiv preprint arXiv:2209.14201, 2022.
[121] L. Du, X. Ye, X. Tan, E. Johns, B. Chen, E. Ding, X. Xue, and J. Feng, “Ago-net: Association-guided 3d point cloud object detection network,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.
[122] X. Weng, J. Wang, D. Held, and K. Kitani, “3d multi-object tracking: A baseline and new evaluation metrics,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 10 359–10 366.
[123] N. Benbarka, J. Schro¨der, and A. Zell, “Score refinement for confidence- based 3d multi-object tracking,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, pp. 8083–8090.
[124] Z. Pang, Z. Li, and N. Wang, “Simpletrack: Understanding and rethinking 3d multi-object tracking,” arXiv preprint arXiv:2111.09621, 2021.
[125] F. Sezgin, D. Vriesman, D. Steinhause, R. Brandmeier and T. Lugner, “Safe Autonomous Driving in Adverse Weather: Sensor Evaluation and Performance Monitoring,” in IEEE Intelligent Vehicles Symposium (IV), 2023.
[126] B. Tefft, “The Prevalence of Motor Vehicle Crashes Involving Road Debris, United States, 2011-2014 (Technical Report),” Washington, D.C.: AAA Foundation for Traffic Safety., 2016.
[127] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in kdd, 1996.
[128] “ISO 23150:2023 Road vehicles — Data communication between sensors and data fusion unit for automated driving functions — Logical interface”.
[129] O. Schumann, C. Wöhler, M. Hahn and J. Dickmann, “Comparison of random forest and long short-term memory network performances in classification tasks using radar,” in 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2017.
[130] N. Scheiner, F. Kraus, N. Appenrodt, J. Dickmann and B. Sick, “Object detection for automotive radar point clouds - a comparison,” AI Perspectives, vol. 3, no. 1, pp. 1--23, 2021.
[131] E. Arnold, et al., "A survey on 3d object detection methods for autonomous driving applications." IEEE Trans. on Intelligent Transportation Systems, 2019.
[132] H. Y. Yatbaz, M. Dianati, and R. Woodman, "Introspection of dnn-based perception functions in automated driving systems: State-of-the-art and open research challenges," IEEE Trans. on Intelligent Transportation Systems, 2023.
[133] D. Miller, et. al., "Dropout sampling for robust object detection in open-set conditions," IEEE Conference on Robotics and Automation (ICRA), 2018.
[134] D. Hu, et al., "Transnet: Transformer-enhanced residual-error alternative suppression network for mri reconstruction," IEEE Trans. on Instrumentation and Measurement, 2022.
[135] G. Melotti, et al., "Reducing overconfidence predictions in autonomous driving perception," IEEE Access, 2022.
[136] W. Miller, "Uncertainty in estimated glomerular filtration rate is much larger than the race adjustment term," Clinical Chemistry, 2021.
[137] M. Ramanagopal, et. al., "Failing to Learn: Autonomously identifying perception failures for self-driving cars," IEEE Robotics and Automation Letters, 2018.
[138] C. Guru, C. H. Tong, and I. Posner, "Fit for purpose? Predicting perception performance based on past experience," Int. Symposium on Experimental Robotics, 2016.

[139] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Per-frame map prediction for continuous performance monitoring of object detection during deployment," IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW) 2021.
[140] H. Y. Yatbaz, M. Dianati, K. Koufos, and R. Woodman, "Run-time introspection of 2D object detection in automated driving systems using learning representations," IEEE Trans. on Intelligent Vehicles, 2024.
[141] H. Y. Yatbaz, M. Dianati, K. Koufos, and R. Woodman, "Introspection of 2d object detection using processed neural activation patterns in automated driving systems," IEEE/CVF International Conference on Computer Vision, 2023.
[142] https://github.com/autowarefoundation/autoware.universe/tree/v0.8.0
[143] M. B. Muhammad and M. Yeasin, "Eigen-cam: Class activation map using principal components," Int. Joint Conference on Neural Networks (IJCNN), 2020.
[144] H. Ramaswamy et al., "Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization," IEEE/CVF, pp. 983-991, 2020.
[145] M. Rezaei, T. Mutsuhiro, and R. Klette. "Robust vehicle detection and distance estimation under challenging lighting conditions," IEEE Trans. on Intelligent Transportation Systems, 2015.
[146] V. Nguyen, et al., "Toward real-time vehicle detection using stereo vision and an evolutionary algorithm," IEEE Vehicular Technology Conference (VTC Spring), 2012.
[147] A. Haselhoff, A. Kummert, and G. Schneider, "Radar-vision fusion for vehicle detection by means of improved haar-like feature and adaboost approach," IEEE European Signal Processing Conference, 2007.
[148] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," IEEE/CVF conference on computer vision and pattern recognition, 2021.
[149] A. Palffy, E. Pool, S. Baratam, J. F. P. Kooij και D. M. Gavrila, «Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset,» IEEE Robotics and Automation Letters, τόμ. 7, αρ. 2, pp. 4691-4968, 2022.
[150] F. Ding, A. Palffy, D. M. Gavrila και C. X. Lu, «Hidden Gems: 4D Radar Scene Flow Learning Using Cross Modal Supervision,» σε Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.
[151] J. Lombacher, M. Hahn, J. Dickmann και C. Wöhler, «Potential of radar for static object classification using deep learning methods,» σε IEEE MTT-S International Conference on Microwaves for Intelligent Mobility, 2016.
[152] A. Palffy, J. Dong, J. F. P. Kooij και D. M. Gavrila, «CNN Based Road User Detection Using the 3D Radar Cube,» IEEE Robotics and Automation Letters, τόμ. 5, αρ. 2, pp. 1263-1270, 2020.
[153] O. Schumann, M. Hahn, J. Dickmann και C. Wöhler, «Comparison of random forest and long short-term memory network performances in classification tasks using radar,» σε Sensor Data Fusion: Trends, Solutions, Applications, 2017.
[154] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley και C. Stachniss, «KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way,» σε EEE Robotics and Automation Letters, 2023.
[155] A. Palffy, J. Kooij και D. Gavrila, «Detecting darting out pedestrians with occlusion aware sensor fusion of radar and stereo camera,» IEEE Transactions on Intelligent Vehicles, τόμ. 8, αρ. 2, pp. 1459-1472, 2023.
[156] H.J.H. Boekema, B.K.W. Martens, J.F.P. Kooij and D.M. Gavrila. Multi-Class Trajectory Prediction in Urban Traffic Using the View-of-Delft Prediction Dataset. IEEE Robotics and Automation Letters, vol. 9 no.5, DOI: 10.1109/LRA.2024.3385693, 2024.

[157] EVENTS. (2024). Deliverable D5.1: System integration in the virtual testing setup (submitted but not yet accepted by the EC).