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

Grant Agreement Number: 101069614

D.3.1: Perception Components Methods

Document Identification			
Status	Final	**Due Date**	31-12-2023
Version	1.0	**Submission Date**	30-12-2023
Related WP	WP3	**Document Reference**	D3.1
Related Deliverable(s)		**Dissemination Level**	PUB
Lead Participant	TUD	**Document Type:**	OTHER
Contributors	All WP3 partners	**Lead Authors**	Dariu Gavrila, TUD
		Reviewers	Bill Roungas, ICCS
			David Fernandez, APTIV

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**
Alireza	Ahrabian	HIT-UK
Kirsty	Aquilina	APTIV
Anastasia	Bolovinou	ICCS
Markos	Antonopoulos	ICCS
Giorgos	Hatzipavlis	ICCS
Leonardo	González Alarcón	TECNALIA
Ted	De Vries Lentsch	TUD
Thomas	Griebel	UULM
Andras	Palffy	PERCIV

Document History			
Version	**Date**	**Modified by**	**Modification reason**
0.1	30/10/2023	TUD	Empty document with section structure
0.2	27/11/2023	All	First overall Deliverable draft, after input from WP3 task leaders. Does not contain Executive Summary and Conclusions.
0.3	13/12/2023	TUD, HIT, ICCS, UULM	New overall Deliverable draft, including Executive Summary and Conclusions.
0.4	20/12/2023	ICCS	Document was reviewed by ICCS
0.5	22/12/2023	APTIV	Document was reviewed by APTIV
0.6	27/12/2023	TUD	Reviewers' comments/suggestions addressed by TUD
0.7	28/12/2023	ICCS	Final review by the project's coordinator
1.0	29/12/2023	ICCS	Final version submitted in the EC portal

Quality Control		
Role	**Who (Partner short name)**	**Approval Date**
Deliverable leader	Dariu Gavrila (TUD)	28/12/2023
Quality manager	Panagiotis Lytrivis (ICCS)	28/12/2023
Project Coordinator	Angelos Amditis (ICCS)	29/12/2023

Executive Summary

Work Package (WP) 3 addresses the development of the environment perception system and its self-assessment within the EVENTS project. The environment perception system involves on-board sensing (using camera, radar, LiDAR), supported by localization technology (GNSS+INS) and HD digital maps, and is potentially augmented by V2X technologies. WP3 provides the algorithmic content of the perception modules described in EVENTS architecture (see Deliverable 2.2 [36]) to address the set of challenging driving scenarios, called experiments (EXP1-EXP8), specified in Deliverable 2.1 [35].

Specifically, Deliverable (D) 3.1 reports on work in progress within Tasks (T) 3.1 – 3.4 of WP3 at month 16th of the EVENTS project (the final status will be reported in D3.2 at month 24th). T3.1 involves the acquisition and adaptation of training data needed for the machine learning-based approaches. A variety of existing public datasets for vehicle-based environment perception were explored. Work is listed on the harmonization of annotations across datasets, allowing unified access and ease-of-use. For those experiments, where no suitable datasets existed, work on sensor placement and calibration is reported. A novel road debris dataset was collected. This Deliverable reports on various data augmentation techniques and on the use of unsupervised learning.

For T3.2, on the topic of semantic scene analysis and precise localization, both novel sensors and algorithms are leveraged to overcome the challenges presented by the experiments. The novel sensors include 4D radars to semantically perceive the vehicles surroundings as well as to support localization in poor weather conditions. Furthermore, there is also a focus on developing methods that can overcome challenges that arise under degraded GNSS conditions using LiDAR-based SLAM approaches.

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. This Deliverable reports on the prediction of vehicle movements at a roundabout. It also covers an environment model for pedestrians, a flexible and adaptive grid map representation of the environment, and an advanced Labeled Multi-Bernoulli Filter. Related to roadworks, unmarked lanes, narrow roads and jammed highways, this task studies the use of Kalman Filter-based state estimatiors.

T3.4, on the topic of augmented perception by V2X, extends the on-board perception of the ego-CAV with information coming from other CCAVs or infrastructure sensors and addresses intersection crossing and roundabout urban scenarios. The information exchange between the vehicle and the external sensors is based on the ETSI standard.

In this context, an experiment that bridges both WP3 and WP4 work is studied, i.e. a coordinated platooning maneuver at a roundabout, where the focus from WP3 is on collective perception techniques by deploying a Bayesian late fusion scheme in the presence of occlusions and sensor measurement uncertainties. First steps are done in simulation, using the CARLA simulator and ROS2 Humble.

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 Automated Vehicle
CP	Collective Perception
CPM	Collective Perception Messages
Dx.y	Deliverable x.y
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
ODD	Operational Design Domain
PERCIV	Perciv.AI (EVENTS project partner)
REQs	Requirements
SAE	Society of Automotive Engineers
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)
VRU	Vulnerable Road User

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. ODD limit is reached due to traffic changes, harsh weather/light conditions, imperfect data, sensor/communication failures, etc.), 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 in order 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 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 its self-assessment. The perception system consists of on-board perception (using camera, radar and LiDAR sensors), which is supported by localization (using GNSS and INS) and 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 EVENT use cases.
- Development of solutions for robust perception in complex urban traffic and urban area parks, which involves a less structured road layout (e.g. unclear/non-existent road markings, narrow roads, bridges), which might be cluttered (e.g. infrastructure like traffic poles, lights and signs, or due to parked cars), and involves close encounters with (possibly multiple) road users from various directions (in particular with VRUs).
- Addressing the challenges of perception in poor visibility conditions due to lighting (e.g. night-time, blinding low-standing sun) or adverse weather (e.g. rain, snow, fog) as well as other sensor impairments.
- Developing techniques for augmenting the on-board perception by using V2X information (e.g. CAM, CPM messages) from the infrastructure and/or from other vehicles.
- Development of methods for self-assessment of perception systems that are able to detect deviations from the intended acceptable performance which can stem from various reasons such sensor impairments and noise, sensor de-calibration, faults in components, as well as errors that may result from 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 localisation (HIT)
- Task 3.3 Environment state estimation and motion prediction (TUD)
- Task 3.4 Augmented perception by V2X (UULM)
- Task 3.5 Perception system self-assessment (WMG)

WP3 includes the in-lab technical evaluation of the algorithmic components developed in T3.2-T3.5.

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. Deliverable D3.1 (this document) covers datasets and describes methods for perception components, as they relate to Tasks T3.1 – T3.4. It also describes evaluation methods

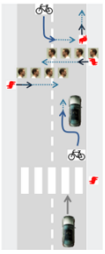
and provides some initial results. D3.1 provides an *intermediate* snapshot of the work done in T3.1 – T3.4. The subsequent Deliverable D3.2, will describe the *final* outcome of WP3, with emphasis on overall perception system and its validation. It will also include T3.5 activities.

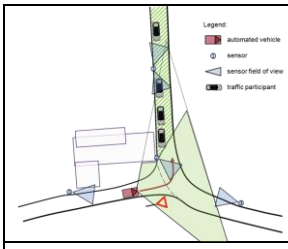
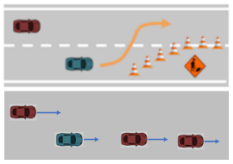
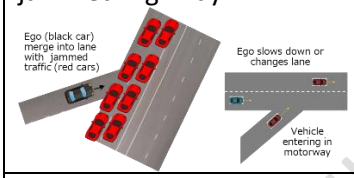
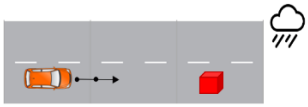
The remaining main sections of the document correspond to T3.1 – T3.4. The following sub-section recaps the EVENTS experiments.

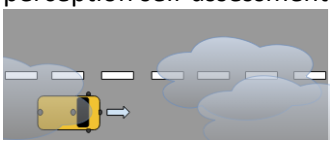
1.3 Experiments

Table 1 recaps the experiments that were selected for demonstration by the EVENTS consortium, as specified in Deliverable D2.1 [156]. 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 (T4.1), 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 cooperative awareness messages (CAM) and collective perception messages (CPM) (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 self-assessment 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 self-assessment 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, WMG 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 SPAT messages, optional, if available), thus increasing the reliability of its position in case of a failure or sensor blockage (T3.4). Particularly in the context of roadworks, unmarked lanes and narrow roads, the ego-vehicle performs a self-assessment by deciding whether to trust its perception system (T3.5).
EXP5 – HIT-FR/UK, CRF, TECN Decision making for motion planning when entering a jammed highway 	EXP5 is like EXP4 with two main differences. The first is that there is not self-assessment (T3.5) of the ego-vehicle. The second difference 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, over-drivability and estimation of time to collision) within diverse weather conditions where the object might not be clearly visible to the human eye and a critical decision on the vehicle behaviors shall be taken to either avoid a potential frontal collision if the object is not over-drivable by braking or avoid a potential rear collision with other vehicles driving behind if the object is over-drivable 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	This experiment focuses on the development of an integrity monitoring mechanism for estimating the distance to the leading vehicle in urban and highway environments under adverse operational domain 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 and tracking becomes unreliable. Another objective (not related with the self-assessment objective) is to study the effects of

conditions 	adverse weather conditions on a perception module performing other vehicles' behavior prediction. *Note that EXP7 will not be covered in this Deliverable, as it involves work performed in T3.5. It will be addressed in D3.2.*
EXP8 - PERCIV Driving minor road under adverse weather conditions including perception self-assessment 	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 therefore on perception and localization of automated driving system. Use case provides 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

2.1 Introduction

Machine learning-enabled perception systems (e.g. deep learning) rely on large annotated training sets to achieve the high performance needed in the context of automated driving (e.g. [16]). However, a supervised learning approach relying on manual annotations does not scale up well; manual annotations are costly and the law of diminishing returns seems to apply almost universally. Further improving the performance of a system for a few more "last percentage points" requires disproportionally larger effort. At the same time, testing conditions are often different to training conditions, because sensors or environmental conditions have changed. This would potentially require the costly annotation effort to be repeated.

Aiming to tackle and alleviate the aforementioned challenges, task T3.1 covers the acquisition and adaptation of training data needed for machine learning-enabled perception systems that would enable us to address the EVENTS use cases. The work has involved various directions:

i. Exploration of existing public datasets (Section 2.2)

ii. Dataset acquisition: sensor placement (Section 2.3)

iii. Acquisition of a new road debris dataset within EVENTS (Section 2.4)

iv. Obtaining an overview of data-efficient techniques (Section 2.5)

v. Data generation and augmentation (Section 2.6)

vi. Self-supervised learning (Section 2.7)

2.2 Existing public datasets

2.2.1 Overview

A variety of public datasets have been explored by the partners involved in task T3.1 (Training data acquisition and adaptation) for vehicle-based object detection and motion prediction.

Figure 1 provides an overview of datasets explored by the involved partners. The exploration and analysis of the datasets in lines 1-15 was carried out in view of the task of VRU detection, tracking and motion prediction, as part of EXP1 (cf. Section 1.3). Recent automotive datasets for 3D object detection use LiDAR-based annotations [2][3][4][7]. Aside from [7], which focuses on radar applications, the datasets offer a significantly higher number of objects compared to previous work [1]. Annotations are provided in the form of 3D bounding boxes [2][4], while [3] and [7] also offer corresponding 2D information. The image-based BDD100k [15] comes at a comparable size, containing solely 2D annotations. Other tracking datasets outside the automotive context have been recently published. First, the STCrowd [17] dataset focuses on person crowds and offers 2D and 3D annotations for synchronized LiDAR and image data. Second, the PersonPath22 [20] dataset contains images mainly captured by static cameras with 2D bounding box annotations.

In contrast, intention prediction focused datasets offer additional attributes (e.g., action, attributes, crossing indication) accompanying the 2D bounding boxes [9][10][11][13][14][21] or 3D annotations [6][12]. Most datasets have been recorded in one or few different cities, offering only limited geographical coverage. Notable exceptions are STIP [14] recorded in six different cities in two different U.S. states, JAAD [10] with five cities in four different countries, and lastly CityWalks [21]. The latter includes recordings from 21 different European cities, by using non-automotive videos from YouTube.

The methods used to further analyze the abovementioned datasets are described in Section 2.7.

	Dataset	#Frames	#Object Annotations	#Unique Objects	Duration (min)	Geographical Coverage	Annotations	Annotation Process
1	ETH Pedestrian	2.3k	18k	Not Specified	3 (~13 FPS)	Zurich, Switzerland	2D BB	Manual Annot.
2	KITTI	7.9k	14k	0.2k	13	Karlsruhe	2D BB, 3D BB	Manual Annot.
3	View-of-Delft	8.7k	38k	0.6k	18	Delft	2D BB, 3D BB	Manual Annot.
4	MOT17	11k	293k	1.3k	8	Not Specified	2D BB	Manual Annot.
5	Argoverse	350k	143k	1.5k	60	Miami, Pittsburg	3D BB	Manual Annot.
6	PIE	294k	739k	1.8k	360	Toronto	2D BB, Crossing, Attributes	Not Specified
7	JAAD	75k	391k	2.8k	46	5 cities, 4 countries	2D BB, Crossing, Attributes	Not Specified
8	nuScenes	40k	246k	4.3k	330	Boston,Singapore		Manual Annot.
9	ROAD	122k	295k	5.0k	170	Oxford	2D BB, Crossing, Attributes	Manual Annot.
10	EUROPVI	83k	299k	7.8k	140	Brussels, Leuven	2D segmentation, 3D locations	Manual Annot.
11	TITAN	75k	499k	<14k	175	Tokyo	2D BB, Attributes, Actions, Crossing	Manual Annot.
12	STIP	1108k	3500k	25k	923	8 cities, 2 US states	2D BB, Crossing	Manual Annot. Semi Supervised
13	BDD100k	400k	509k	27.5k	1333	4 areas, US	2D BB	Manual Annot.
14	WOD	401k	2863k (3D) 2237k (2D)	23.9k (3D) 46.6k (2D)	676	6 cities, US	2D BB, 3D BB	Manual Annot.
15	ECP2.0	2057k	8762k	277k	1714	29 EU cities, 11 countries	2D BB, 3D locations	Manual Annot. Semi Supervised
16	Cityscapes	4k	103k	103k	NA	50 cities	2D BB (extracted), Segm. Masks	Manual Annot.
17	Coco	82k	385k	385k	NA	Not Specified	2D BB (extracted), Segm. Masks	Manual Annot.
18	LISA	36k	110k	Not Specified	~24	San Diego, California, USA	2D BB	Manual Annot.
19	Mapillary Vistas	20k	635k	Not Specified	~13	Cities in all 6 continents	2D BB (extracted), Segm. Masks	Manual Annot.
20	Open Images V4	368k	1,230k	1,230k	NA	Not Specified	2D BB	Manual Annot.
21	Udacity	13k	93k	93k	NA	Not Specified	2d BB	Manual Annot.

Figure 1: Overview of explored datasets

Datasets in lines 1,8,13 and 16-21 were further explored with a broader perspective on vision-based perception and prediction. The methods, which are going to be further processed and utilized, are described in detail in Section 2.6. A total of 15 classes of interest have been defined by taking into consideration all pertinent annotations in each dataset. The resulting object class statistics are provided in Figure 2.

	Class	person	bicycle	car	motorcycle	rider	bus	train	truck	van	traffic light	fire hydrant	pole	traffic sign	cat	dog
Dataset	**# frames**															
BDD100k	79863	104611	8217	815717	3454	5166	13269	151	34216		213002			274594		
cityScapes	3471	21413	4904	31822	888	2363	483		582		11898			24976		
COCO	82124	273469	7429	45799	9096		6354	4761	10388		13521	1966		2058	4970	5726
LISA	36265										109475					
Mapillary V.	19780	58181	10886	147370	6239	4458	4858	271	7538		77038	2226	316069			
nuScenes	154235	233686	10030	582159	9526		18261		106274	57						
Open Images v4	367768	824221	36763	253583	12917	0	11372	11821	12386	8349	7346	500	0	6150	14677	32327
ETH Pedestrian	2387	17779														
Udacity	13063	9866		60788		1676			3503		17253					
SUMS	758956	1543226	78229	1937238	42120	13663	54597	17004	174887	8406	449533	4692	316069	307778	19647	38053

Figure 2: Object presence statistics for classes of interest.

Additionally, the PREVENTS [122] and ROAD [121] datasets have been reviewed in quest of lane changing scenarios that could support ICCS work in T4.2 (other road users' behavior prediction).

Finally, a review of recently released real-world captured datasets that can be used for Collective Perception research (incl. infrastructure data) was performed: Several surveys of highly qualified content have been released in the last two years trying to cover the CP task research and development challenges, proving the growing interest of both AD perception and VANETs communities [118][119][120]. DAIR-V2X, OpenDAIR-V2X and soon to be released V2X-Seq [15] are the first real-world datasets for research on V2X-enabled Collective Perception. It comprises image data and LiDAR point cloud data from different observers. Notably, it supports early fusion and late fusion CP methods while it is planned to also support feature fusion. For more details on CP datasets created via simulation, refer to Section 2.6.2.

Popular public datasets like nuImages, COCO 2017, Berkeley DeepDrive and KITTI do not have class annotations for traffic signs, especially non-standard ones. Public datasets containing rich traffic sign annotations are Mapillary [18] and Zenseact [19].

Mapillary contains annotated traffic signs only (cars, pedestrians are not annotated). There are 14K training images, 5K validation images. High resolution dash camera images are the predominant data source. There are 330 traffic sign classes, with approximately 260K sign labels. Mapillary is used by project partner HIT in Section 2.3.

In the context of public motion prediction datasets, there are several options available for research. A comparison of different motion prediction datasets is shown in Table 2. These datasets provide past trajectories of agents around the ego-vehicle to calculate their future trajectories. In addition, some of them have vector maps to account for road factors in the model.

Table 2: Motion prediction datasets.

	Argoverse 1 [215]	Inter [216]	Lyft [217]	Waymo [218]	nuScenes [219]	Yandex [220]	Argoverse 2 [221]
Scenarios	324k	-	17k	104k	41k	600k	250k
Unique tracks	11.7M	40k	53.4M	7.6M	-	17.4M	13.9M
Average route length (s)	2.48	19.8	1.8	7.04	-	-	5.16
Total time (h)	320	16.5	1118	574	5.5	1667	763
Scenario duration (s)	5	-	25	9.1	8	10	11
Forecast horizon (s)	3	3	5	8	6	8	6
Sampling rate (Hz)	10	10	10	10	2	5	10
Cities	2	6	1	6	2	6	6
Unique roadways (km)	290	2	10	1750		-	2220
Average track per scenario	50	-	79	-	75	29	73
Evaluated object categories	1	1	3	3	1	2	5
Multi-agent evaluation	-	X	X	X	-	X	X
Vector map	X	-	-	X	X	-	X
Size (GB)	4.8	-	22	1400	45	120	32

The most utilized datasets for motion forecasting are nuScenes [219], Waymo [218], Argoverse 1 [215] and Argoverse 2 [221]. For this reason and due to the supportive open-source community surrounding Argoverse 1, APTIV's intention is to utilize it for training purposes (cf. Section 4.2.3).

2.2.2 Harmonization of annotations

A variety of publicly available image datasets has been explored, processed and analyzed by ICCS, as already shown in previous Subsection, targeting the following outcomes:

1. To obtain object class statistics of each dataset special attention has been given to classes pertaining to VRUs (e.g. pedestrians, cyclists). Based on the prevalence of each object in each dataset, a total of 15 classes of interest have been defined taking into consideration all pertinent annotations in each dataset. The results of this work are depicted in Figure 2. The 15 object classes of interest are shown in the first row (e.g. person, cyclist, dog). Numbers in the following rows show the number of (annotated) cases of each object class contained in each corresponding dataset (first column in Figure 2).

2. To transform the annotations of each dataset according to a single, homogeneous and flexible annotation template common for all images across all datasets. To this end, a custom JSON file template has been designed, an example of which is shown in Figure 3.

```
{
    "image_id": "cityScapes_train_1.jpg",
    "image_path": "data/cityScapes/train/cityScapes_train_1.jpg",
    "width": 2048,
    "height": 1024,
    "annotations": [
        {
            "class": "car",
            "bbox": {
                "x1": 609,
                "y1": 420,
                "x2": 807,
                "y2": 532
            },
        },
        {
            "class": "car",
            "bbox": {
                "x1": 145,
                "y1": 429,
                "x2": 304,
                "y2": 502
            },
        }
    ]
}
```

Figure 3: Annotation template (json).

Each image is accompanied by a corresponding JSON file of the depicted format, containing information on the name, path and dimensions of the image, plus a list of annotations (object class, bounding box coordinates) of each object of interest present in the image.

The nuScenes dataset mentioned above contains both LiDAR and radar data, see Figure 4. Scripts for acquiring and depicting such data have been developed, however, publicly available LiDAR and radar data are much less common than camera images. The SeeingThroughFog [22], RADIATE [23] and CADC [24] datasets are currently under exploration by ICCS and APTIV.

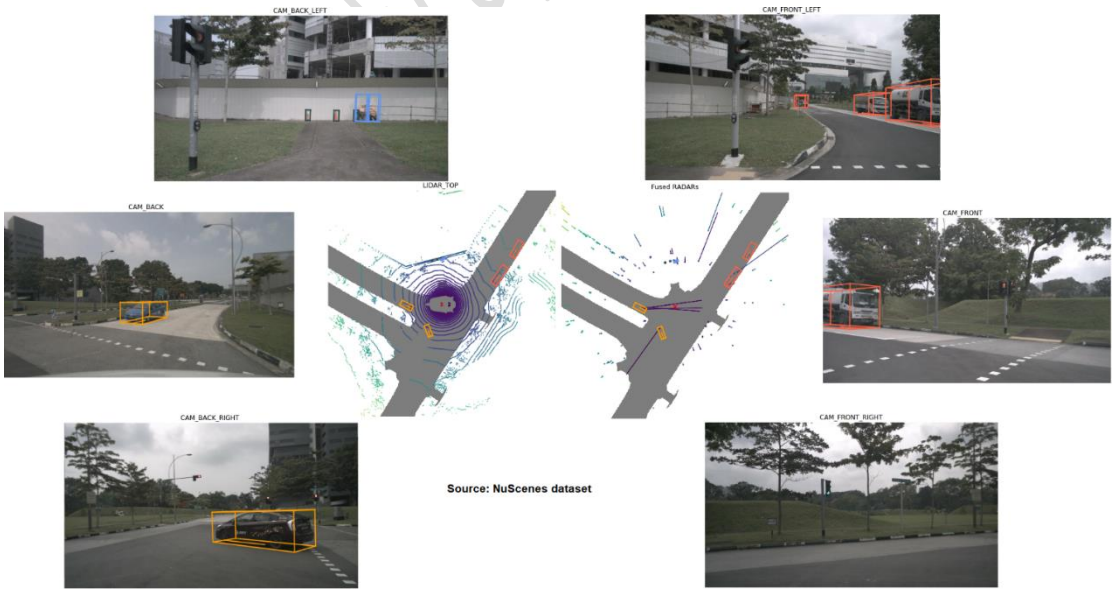


Figure 4: nuScenes dataset. Two central images; left depicts LiDAR data, right depicts radar data. Images on the left depict left, center and right back cameras. Images on the right depict left, center and right front cameras.

2.3 Dataset acquisition: sensor placement

This subsection describes the sensor set and the strategy to identify the optimal placement of those sensors for the activity of data acquisition by partner HIT in the project.

Figure 5 shows the design of sensor suite on HIT's demo car for EVENTS. The sensor suite and car are designed and simulated in Gazebo software [25]. The sensors for the perception system include two 360° Ouster LiDARs and four mono cameras. The specifications of cameras and LiDAR are as follows:

- ✓ **Cameras**: Sony IMX390 CMOS sensor camera; 1920x1080 @30fps, HFOV 120.6°
- ✓ **LiDAR**: Ouster OS1-128 (rev7), 10-20fps, HFOV 360°, VFOV 45°, range 0.5-200m, Vertical Resolution 64 or 128, Horizontal resolution 512, 1024 or 2048

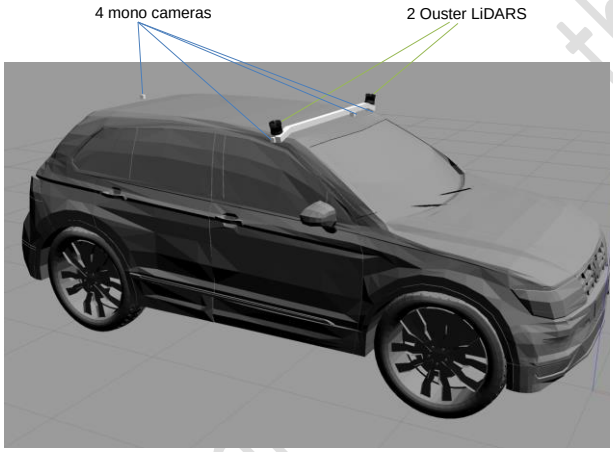


Figure 5: Sensor setup on Tiguan – HIT demo car in simulator

To evaluate the coverage of the sensor setup, Gazebo and ROS Rviz [26] software has been used to simulate the output of the camera and LiDAR point cloud data. Figure 6 shows the coverage evaluation of the output from cameras and LiDARs. By using the simulator to simulate the output of the sensors, the optimal sensor suite can be easily found (number of sensors, mounting position, mounting angle, …) before setting it up on the real prototype vehicle.

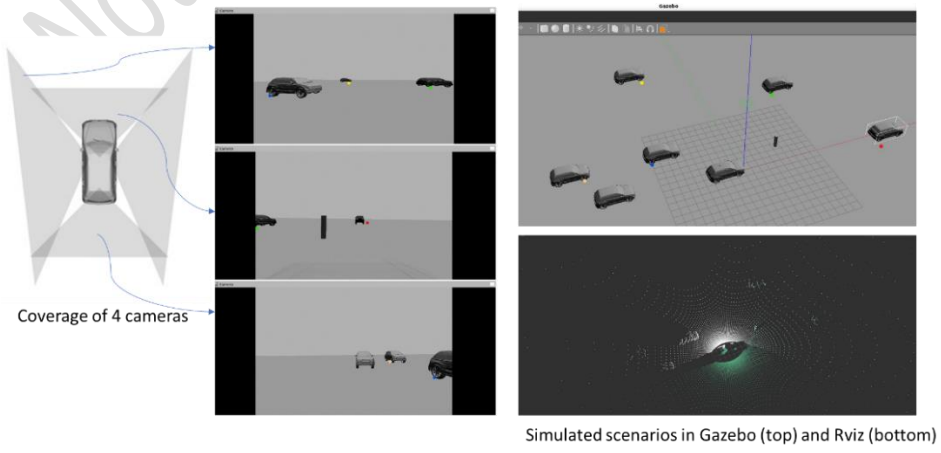


Figure 6 Coverage evaluation of the sensor suite

2.4 Dataset acquisition: new road debris dataset

This section describes APTIV's activities to obtain the training data required for its perception module to support EXP 6 ("Small object detection at a far range in adverse weather conditions", cf. Table 1). In Section 2.4.1, the aim is to understand the experiment in more detail and subsequently use that information to design the training data collection procedure. Additionally, it is important to identify which characteristics of the small objects may affect the driving task, not only from a safety perspective but also from a driving comfort perspective. In Sections 2.4.2 and 2.4.3, two major categories related to objects that can create discomfort to the driver and passengers are considered. Finally, in Sections 2.4.4 and 2.4.5, the data collection procedure, a proposed overdriveable height threshold and the plans for future data collection for the perception evaluation are presented.

2.4.1 Debris accidents

Looking at the Traffic Safety Facts Annual Report Tables from NHTSA [37], it has been found that 1.4% [38] of the 5,251,006 accidents in 2020 were due to collisions with non-fixed objects. These 74,430 accidents are directly related to Experiment 6. During 2020 the number of vehicles miles travelled was 2904 billion which would lead to 2.2 accidents every 100 million km driven.

Additionally, the latest key figures from the European Commission road safety website [39] show that most traffic fatalities for car occupants occur when no other vehicle is involved [40]. Therefore, automated vehicle systems should not only focus on preventing collisions with other vehicles but also focus on avoiding collisions with static elements on the road.

According to an article about lawsuits [41], debris can be defined as any object that should not be on the road. A lot of debris can be items that have fallen off trucks such as **branches** that were meant for landscaping, **scrap metal, wood or building material, bales of hay, fruits and vegetables** (from produce trucks). Additionally, debris might also be objects which have not been secured correctly by passenger cars such as **furniture, lumber, luggage and rubbish**. For instance, rubbish bags/plastic bags filled with clothes might be a good example. Furthermore, according to another article [42], debris might also include **mufflers, bumpers, hubcaps, light poles, traffic signs, construction cones or barrels and railroad ties**.

The AAA Foundation for Traffic Safety [43] produced a report [44] about debris-related accidents by analyzing data from NHTSA (NASS CDS, GES and FARS). The report estimated a total of 50,658 accidents per year between 2011 and 2014 with an average of 9,805 injuries and 125 deaths each year. An accident was considered to be debris-related if it involved: a collision with an object falling from another vehicle, a

collision with a non-fixed object, or an accident due to avoiding collision with a non-fixed object. This is a list of debris captured in the study:

- vehicle part debris (wheels, tires and other vehicle parts such as vehicle jack, tire rim, tire thread, driveshaft)
- furniture (sofa, chair)
- appliances
- detached trailer
- fallen trees, branches, limbs
- rocks and boulders
- poles, steel beams, metal ramps, plastic barrels, construction barrels, boxes, garbage cans

The report also pointed out that debris-related crashes were more likely to result in property damage than injury or death. According to an article [45] published by a lawyer on his company's website, examples of such damage include dents, punctures and engine and transmission damage amongst many others.

A few recent examples of documented debris accidents are shown in Table 3. These were obtained by searching the Crash Investigation Sampling System (CISS) database [46]. Only the impact with the debris in question was considered but the accident could have led to other subsequent events which could have influenced the severity ranking. For a full description of the accident refer to the URL provided.

Table 3: Examples of debris accidents from CISS.

Year	Notes	Severity	URL
2019	Impact with plastic barrel	Moderate	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=15298#
2019	Damage with wooden block	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=14019#
2019	Impact with a ladder laying on the road	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=15369#
2019	Impact with a wheel	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=12575#

2019	Impact with a tree branch	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=12655#
2020	Impact with a rock. Dimensions are about 42 cm diameter and 33 cm height	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=19800
2020	Impact with a light pole laying across the street	Light/Unknown	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=18930#
2020	Impact with a temporary construction sign	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=18485#
2020	Impact with a boulder	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=16764#
2020	Impact with a mattress	Unknown	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=17068#
2020	Impact with a construction pylon	Unknown	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=17069#
2020	Impact with metal debris	Unknown	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=20234#
2021	Impact with a metal object with dimensions 79 cm (length) by 15 cm (width) by 15 cm (height)	Severe	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=20476#
2021	Collision with a rock that damaged the undercarriage	Moderate	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=22979#
2021	Impact with a fallen tree	Moderate	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=22323#
2021	Damage due to a fallen tree	Moderate	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=23369#
2021	Damage due to impact with a tire	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=22240#
2021	Impact with pile of dirt	Light	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=22811#

2021	Impact with a wood pallet	Light/Unknown	https://crashviewer.nhtsa.dot.gov/CISS/Details?Study=CISS&CaseId=23232#

These recent accident examples agree with the objects mentioned in this literature survey and provide some examples of debris dimensions. Therefore, it helps us understand better the use case and highlights the importance of having a system that can reduce such accidents.

2.4.2 Driving on rough terrain

Understanding what kind of objects are typically road debris can be complemented by understanding what constitutes an object that can be driven over. We will be looking at two branches of literature which can help answer this question: the traversability of rough terrain and the traversability of speed bumps.

In this survey, we want to extract which are the most important parameters to assess the ease of travelling over rough terrain as these parameters might be used for our debris-use case. Traversability is a measure of how easy it is to travel along a certain part of the terrain [47].

Traversability depends on the terrain geometry (slope and roughness) and the interaction of the terrain material with the vehicle [48]. A common approach is to define some weighted average of these attributes as a measure of traversability.

Zhang et al. [47] propose a measure based on terrain attributes (slope, step value and unevenness); a similar metric using roughness instead of unevenness is used in [49]. Zhou et al. [50] also define traversability as a function of roughness and slope but only define it where it is feasible considering the robot's chassis height and the robot's climbing angle. In [51] authors characterise the terrain traversability by using the slope and the step size keeping into consideration the physical characteristics of the robot.

Other studies use vehicle states to estimate the traversability. In [48], they compute a measure based on angular velocities in w_x and w_y and linear acceleration (a_z). In [52], traversability is a function of the vehicle's pitch and roll along with the roughness and height difference of the terrain. In [53], the proposed measure depends on the ground clearance, angular position, angular speed, slope per wheel and distance to the target position; the latter variable is not of interest to the required literature survey.

From these studies we can conclude that the following factors can be used to create a cost function for debris:

1. The slope of the object
2. Unevenness/roughness

3. Vehicle pitch angle
4. Vehicle roll angle
5. Angular speed along the x and y directions (speed of the roll and pitch angle)
6. Linear acceleration in the z direction (acceleration orthogonal to the ground plane)
7. Height of the object compared to the ground
8. Height of the object compared to the vehicle clearance
9. Maximum permissible angles depending on the actual vehicle.

Looking at off-roading articles [54] the following are some of the factors mentioned: approach, departure and breakover angles and ground clearance (the clearance [55] of an on-road vehicle can be between 170 mm and 180 mm whilst that of an off-road car can be between 215.9 mm and 254 mm). These factors agree with the previously mentioned studies. These factors which are vehicle-dependent should be used to derive thresholds for the parameters of interest mentioned above.

These extracted factors from the literature can then be used to find a suitable height threshold for the overdriveability classifier; from above it can be immediately deduced that for on-road vehicles the height threshold for an overdrieveable object has to be smaller than 17 cm. Additionally, these could also be used to help with the decision-making module and lower the velocity of the vehicle to reduce a cost function made of these parameters. This is not something which will be addressed in this project but is provided for completeness' sake.

2.4.3 Speed bumps

Another aspect to consider is speed bumps as they are overdriveable segments of the road and understanding how a vehicle interacts with them can also shed light on our overdriveability scenario.

A speed bump is a raised area on the road ranging between 7.6 cm to 15.2 cm in height and between 15.2 cm and 91.4 cm in width [56]. These are used in low-speed roads and parking lots as they cause discomfort when traversed using the usual residential area's speed ranges, requiring the driver to travel at around 8 km/h [56]. Interestingly, the greatest discomfort is experienced at low-speed ranges where there is peak vertical acceleration and decreases with speed increases owing to the vehicle suspension absorbing the forces before the vehicle reacts to them [57]. However, traversing speed bumps at high speed is not recommended as it can damage the vehicle [58].

Watts [59] shows that vertical acceleration is related to the comfort experienced by people when traversing speed humps; however, the author also points out that other factors such as jerk, horizontal vibration and cognitive stimuli might also influence comfort. Short humps (similar to bumps) produced higher levels of acceleration at slower speeds agreeing with [57]. Even though, at higher speeds, there is less vehicle body motion, the tyres and suspension would still be affected causing vehicle damage and there might be a possibility of losing control of the vehicle.

Complementing the cost function approaches presented in the rough terrain section, ISO 2631 can be used to derive a comfort measure using filtered acceleration values; this approach was used in autonomous vehicle comfort studies [60] and also speed bumps comfort studies [61].

2.4.4 Data collection procedure

Experiment 6 explores a challenging, important but quite niche topic so the required dataset needed to be collected specifically for this task in a controlled environment.

A prototype vehicle equipped with a front-facing radar [62] and a GNSS/IMU system is used to collect data on a test track. The debris is positioned on a straight line marked on the test track as shown in Figure 7.

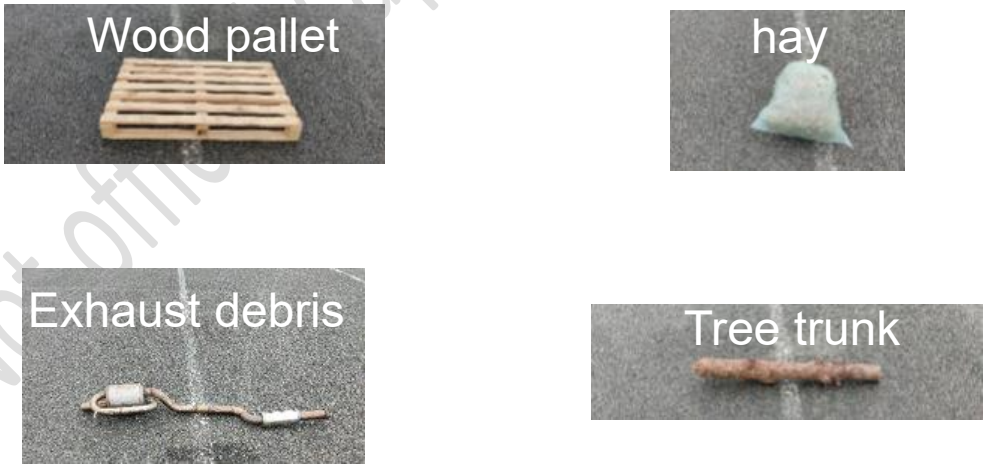


Figure 7: Examples of objects used for data collection. The objects are positioned on the white line painted on the test track.

The vehicle travels for approximately 250 m towards the debris as shown in Figure 8 at two different maximum speeds.

Figure 8: Data collection setup. The vehicle drives in a straight line towards the debris.

A total of 47 different objects were used in the data collection exercise, mostly during dry weather or light-rain conditions. This is a list of the collected objects with their height in brackets:

1. Overdrive plates (1 cm)
2. Oval Metal bar (1.5 cm)
3. Flat metal grid (2 cm)
4. Cylindric metal bar (2 cm)
5. Crowbar (2.5 cm)
6. Metal bars stacked (3cm)
7. Cubic bars (3cm)
8. Flat metal side-by-side (4cm)
9. Flat sign (4 cm)
10. Steel pipe (6cm)
11. Angle rail (6cm)
12. Metal parts (6 cm)
13. Wires (6 cm)
14. Brick (6.5 cm)
15. Traffic panel horizontal (6.5 cm)
16. Pole Support (6.5cm)
17. Metal stack side by side (7 cm)
18. drying rack in a cardboard box (7.5 cm)
19. Bricks side-by-side (7.8 cm)
20. Branch parts side-by-side (8 cm)
21. Flat metal items (8 cm)
22. Metal jack (8.5 cm)
23. Metal ladder flat (9.5 cm)
24. metal bars side-by-side (9.5 cm)
25. Three stones side-by-side (10 cm)
26. Plastic toolbox (10 cm)
27. Woodblocks side-by-side (10 cm)
28. Sprayed Wood (10 cm)
29. Metal cube (10 cm)
30. Soda can (11.5 cm)
31. Sign stand (12 cm)
32. Tool case (12 cm)
33. Wood pallet (13 cm)
34. Metal stack flat (14 cm)
35. Shovel (16 cm)
36. Tree trunk (17 cm)
37. Large stone (20 cm)
38. Exhaust (20 cm)
39. Wheel (22.5 cm)
40. Tire without rim (22.5 cm)
41. Fire extinguisher (25 cm)
42. Plastic cans (26 cm)
43. Helmet (30 cm)
44. Bucket horizontal (31 cm)
45. Bucket vertical (32 cm)
46. Hay bag (36 cm)
47. Metal stack front side (40 cm)
48. Small bicycle (40 cm)
49. Small stair (44 cm)
50. Empty barrel (52 cm)
51. Pylon (54 cm)
52. Table (62 cm)
53. Two pointy poles (75 cm)
54. Wood pallet standing (80 cm)
55. Drying rack standing (90 cm)
56. Metal stack up (100 cm)
57. Metal bar vertical (100 cm)
58. Metal ladder up (138 cm)
59. Sign with stand (300 cm)

Looking at the height for the various objects and our literature review, it was decided to have a cut-off height of **12 cm** for overdriveable objects as items above this height were mentioned in documented accidents. Additionally, from the rest of the literature review, it must be highlighted that overdriveable objects should only be driven over at very low speeds to reduce the damage to the vehicle.

2.4.5 Bad weather data collection (future work)

Data collection for debris from different height groups for different weather conditions and at different times of the day (night/day) will be collected during winter to evaluate the perception module's performance. This will enable us to evaluate the perception performance not only by using a train/test split of the data already collected in mainly good weather conditions but also on an entirely different test dataset collected in bad weather conditions. Dataset collection will be reported in subsequent D3.2.

2.4.6 Data collection for sensor calibration

on simulator.

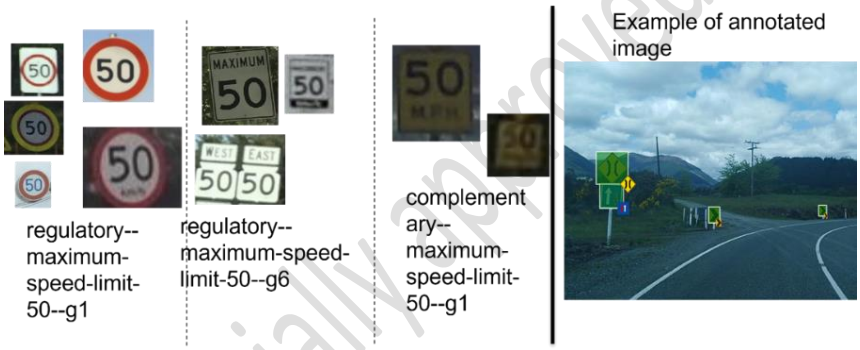


Figure 9: Examples of annotated traffic signs from the Mapillary dataset [18].

2.5 Data efficient techniques – Overview

Data-efficient techniques can be roughly divided into eight categories, namely:

1. **Transfer learning** reuses a model developed for task A as a starting point for a method for task B.
2. **Prior knowledge-based techniques** typically use expert knowledge to improve the performance by easing the learning task.
3. **Data generation and augmentation** either uses simulation to generate synthetic data or augment data based on existing real-world data.
4. **Semi-supervised learning** uses unlabelled data to modify or reprioritize hypotheses obtained from labelled data alone.

5. **Weakly supervised learning** is similar to supervised learning but uses relatively weak labels that are easier to obtain than the labels used for supervised learning.

6. **Active learning** aims to maximize a model's performance gain while annotating the fewest samples possible. The assumption is done that different samples in the same dataset have different values for the update of the current model.

7. **Self-supervised learning (unsupervised learning)** utilizes pseudo labels to learn representations of the data. The pseudo labels are generated automatically based on the attributes found in the data, and after self-supervision, model is finetuned for downstream task.

8. **Cross-modal supervision** fuses the information between two different modalities in order to help training the model, e.g. use LiDAR point cloud to supervise camera-based depth estimation.

2.6 Data generation and augmentation

2.6.1 Real-world data augmentation

Case study: HIT

The public datasets, explored by HIT and listed in Section 2.2, can be grouped as follows: 1) datasets that include only base classes (large number of datasets and a large number of class instances), and 2) datasets that only include traffic signs (a low number of datasets and a low number of class instances) like the example shown in Figure 9 of the Mapillary dataset [18]. This of course limits our ability to re-train a CNN in detecting both base classes and traffic signs; as the public datasets with base classes have a very large number of class instances, whilst the public datasets with traffics signs have a very low number of class instances. To resolve this fundamental imbalance between number of traffic sign class instances and the number of base class instances, we propose to augment existing public datasets to increase the number of traffic sign class instances. We achieve this by developing the following two approaches:

1. Approach I patches an object onto image without considering camera viewpoint and realistic object placement in the scene.
2. Approach II considers the camera viewpoint and object placement in the scene to obtain more realistic annotated dataset.

Approach I steps are as follows:

- Identify a public dataset with a large variation in the scenes. Collect a database of traffic signs to patch onto the public dataset.
- Randomly select a traffic sign to patch onto a single image from the public dataset.
- Randomly resize the selected traffic sign and apply data augmentation techniques (pixel transformations and intensity transformations)
- Identify a random patch on the single image from the public dataset and patch onto the image the transformed traffic sign.

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

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

Approach II considers both the placement of the sign within the scene as well as the position of the camera. The steps for approach II are as follows:

- Identify existing autonomous vehicle dataset. Collect a database of traffic signs to patch onto the public dataset.
- For a given sequence (in the AV dataset), randomly select a traffic sign and location in a global coordinate frame where the sequence data has been collected. This can be carried out multiple times for a given sequence.
- Using projective geometry, project the traffic sign image onto the image plane. This approach considers both the viewpoint of the vehicle camera as well as the configuration of the traffic sign in the sign.

An example of the patch augmented image using the approach II is shown in Figure 11.

Figure 11: Augmentation of traffic signs considering both placement of sign in the scene and the camera viewpoint.

Case study: ICCS

a) Artificially generated image datasets based on an available image dataset

An investigation on methods of generating new images from existing datasets and various methods of augmenting already available images has been carried out. Preliminary outcomes of this investigation are outlined in the following.

1) Throughout the literature, Generative Adversarial Networks (GANs) are a quite common approach in translating/adapting already existing images to required domains (e.g. a sunny image to a cloudy or rainy one like in Figure 12) [27][28][29][30]. The (Multimodal) Unsupervised Image-to-image Translation (MUNIT) approach in [30] is representative of the state of the art in this research area. However, all of these methods are based on GANs, whose training process can be expected to be unstable, unpredictable and time consuming. Furthermore, as indicated in the public repository of MUNIT, the required computational resources are highly intensive in terms of time and hardware, just to provide images of quite low resolution (256x256) [31][32]. Used as a data augmentation tool, an alternative state-of-the-art GAN architecture (pix2pixHD) presented in [33], although less demanding in terms of hardware, produces images of questionable usability in terms of quality, especially in VRUs (Figure 13). The results of [34] (code partially available in the corresponding open-source repository), for augmenting the CityScapes dataset via artificially imposing various adverse weather and light conditions have been successfully reproduced (Figure 14), however, and since the method again relies on GANs, the code is not generic enough to be applied on other datasets. Although the results in adverse CityScape dataset are visually good, the merit of this augmentation technique remains to be proved. In general, the GAN-based approaches studied so far appear not promising in terms of both cost efficiency and usability of the results. ICCS plans to evaluate whether an object detector trained in CityScape and CityScape Adverse performs better in

adverse weather and night-time real world captured data (to be reported in D3.2).

2) Apart from the aforementioned GAN-based approaches, more classical augmentation methods are also under consideration, in terms of evaluating their effect on improving the performance of pertinent machine learning algorithms. These methods include mathematical transformations like perspective and affine transformations, Gaussian blurring plus combinations of morphological filters, color transfer between images plus combinations of all the above. Preliminary experiments show that augmenting image datasets through perspective transformations of existing images and their corresponding annotations may increase the mAP evaluation metric of YOLO object detectors by a few points (up to ~ 0.05) but further experimentation is required.

3) A set of scripts for automated image annotation has been implemented. The scripts are currently based on the YOLO family of object detectors but can be extended to utilize any object detection model (including the R-CNN family) to automatically annotate raw input images. The correctness of the annotations is certainly dependent on the performance of the utilized model. However, using the implemented scripts as a first step is expected to significantly speed up the entire annotation process.

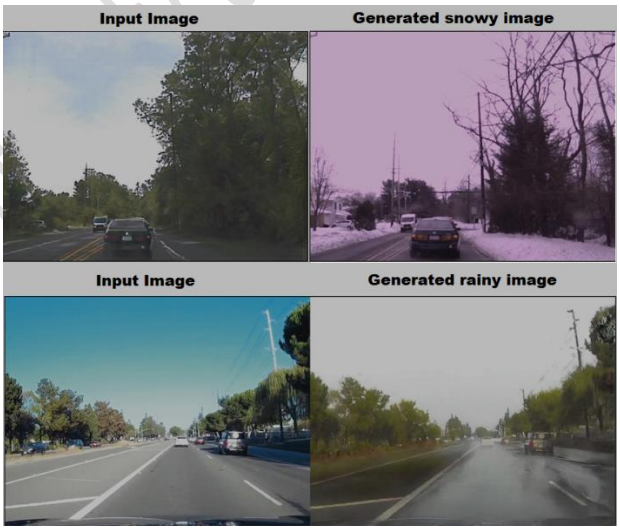


Figure 12: Example of image adaptation using Generative Adversarial Networks (GANs).

Figure 13: An example of questionable usability of VRUs production in simulations.

Figure 14: The augmented CityScapes dataset.

ICCS's work in T3.1 will evolve till D3.2 submission by working on the following camera data augmentation items:

a. Generate adverse weather synthetic data using various deep learning generative model algorithms.

b. Evaluate synthetic data generation as a data augmentation technique by testing a SoA object detector on the augmented CityScape adverse dataset and reporting the results.

2.6.2 Data generation by simulation

This subsection describes work performed by project partner ICCS on data generation in the CARLA simulation environment.

Within the research stream of cooperative connected automated driving, there are many efforts in recent years on introducing large scale Collective Perception (CP) datasets. Due to practical reasons (multi-agent setup in real world or in test tracks is challenging for many reasons), these datasets are mostly generated by simulation. This advent is partially due to the introduction of and community support for CARLA (Car Learning to Act) open source driving simulator [123] which made easy the generation of virtual sensor data and offers a variety of ground truth data (incl. instance semantic segmentation data that yields a unique pixel value for every object in a scene and pedestrian skeleton data). CARLA supports basic sensor modelling for several sensors such as cameras, depth cameras, LiDAR (simulated ray cast), IMU and RADAR. In July 2018, version 0.9.0 introduced the multi-client multi-agent support that opened the road for cooperative agents.

CP datasets generated in simulation: The first big contribution was made in 2021 by UCLA Mobility lab with the release of OpenCDA dataset and simulation benchmark [124] which supported testing both on individual autonomy level and traffic level and built a co-simulation platform (CARLA is part of it), a full-stack prototype cooperative driving system, and a scenario manager. OpenCDA also 35ptimiz benchmark testing scenarios, state-of-the-art benchmark algorithms for all modules of an AD stack, benchmark testing road maps, and benchmark evaluation metrics but focusing on cooperative driving applications and not on CP. It is also noted that type of data exchanged among traffic agents are freely explored and not restricted to the messages already standardized by ETSI. OpenCDA has grown since then and today it constitutes an Open-source Ecosystem for Cooperative Driving Automation Research with an active community developing part of it around the globe [125]. Two similar efforts followed focusing on multi-agent perception: OPV2V [126] and V2X-Sim [127]. Both datasets provide object-level annotations from CARLA towns in order to support detection, tracking and segmentation perception tasks deploying early fusion/feature-fusion/late-fusion approached for CP content generation. OpV2V is implemented by the COOD framework Ih supports multi-GPU training and is used by V2Vset [128]. Scenarios supported by all the simulation frameworks above focus on urban, rural and more seldom on highway operational domain and include V2X in straight and curvy urban road segments, urban intersections, rural areas where only V2V applies and highway on-ramp scenarios. The aforementioned CP simulation benchmarks are

equipped with RGB cameras and Lidar, allowing the collection of more than 10,000 frames and each scene contains at least 2 vehicles. As discussed in V2XviT [128], in simulation different perception error models can be introduced (pose error, agents' synchronization error, time delay in V2X communication).

Regarding the ICCS experiments on collective perception, CARLA (Car Learning to Act) simulator[1] was a good candidate for the environment/sensor/traffic simulation as it binds well with external ROS modules that are needed for CP synthetic falsification data generation. In ICCS CP approach no raw sensor data will be used as late-fusion technique is employed: that means that CARLA sensor groundtruth data will not be recorded but instead, object reference data have to be generated from CARLA for EVENTS experiments. In order to obtain such object reference data, CARLA object semantic segmentation groundtruth can be used and then bounding boxes shall be constructed by us. As shown in Figure 15, in order to build custom CARLA map and scenarios, by importing map data from the real world (in OSM format), Matlab RoadRunner tool is used (see steps 1 and 2 in the figure). Scenario creation using CARLA python API is also not optimal as any change to the scenario shall be hardcoded in the python code. As an alternative more efficient process, Matlab RoadRunner scenario creation is employed and exporting to standardized scenario description format of OpenScenario 1.0 or OpenScenario 2.0 (see step 3 in the figure, where a roundabout scenario is created to be used in T3.4). Once the CARLA scenario is ready to be executed, with one or multiple AV agents inside, the next step is to start replaying the scenario and recording object reference data needed for CP experiment development. Two types of data are needed:

i. Detected object data captured from the Avs perception layer, essentially within each AV's FOV (performed by step 4).

ii. CARLA ground-truth data for the entire scene representing the global knowledge that is missing during a collective perception experiment without an infrastructure camera node (available by CARLA and shown as step 5).

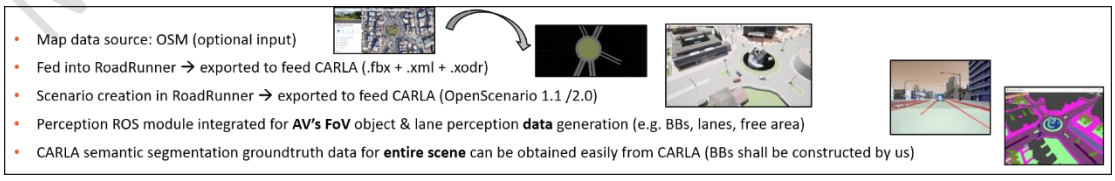


Figure 15: RoadRunner-CARLA data generation pipeline for obtaining object-reference data

[1] https://carla.org/

ICCS's work in T3.1 will evolve till D3.2 submission by working on the following items:

- RoadRunner-CARLA scenario generation pipeline
- CARLA multi-agent sensor and scenario data generation

2.7 Self-supervised learning

This Subsection was contributed by EVENTS partner TUD. After analyzing the various types of data-efficient techniques (cf. Subsection 2.5), self-supervised learning and cross-modal supervision have been pursued as promising techniques as neither needs manual annotations.

Object detection is one of the core tasks of computer vision, and it is part of the pipeline of many applications like face recognition [64], robotics [65], and autonomous driving [76]. During the past decade, the community has made tremendous progress in detecting objects, especially learning-based methods. These methods rely on manual annotations, i.e. the object instances are indicated with a bounding box and labelled with a class identifier. However, a massive amount of training data is usually required for training those models, while labelling is expensive and laborious. This raises the question of how object detection models can be trained without using direct supervision.

Unsupervised object detection is a relatively unexplored research field compared to its supervised counterpart. For camera images, recent work showed that the emergent behaviour of models trained with self-supervised learning could be used for object discovery [68]-[91]. The behaviour implies that the learned features of those models contain information about the semantic segmentation of an image, and thus, they can be used to differentiate between background and foreground. Consequently, the extracted coarse instance masks have been used as a self-supervised signal to train 2D object detectors [94]-[105]. Although these methods perform well for images depicting a few instances, they fail to achieve high performance for images depicting many instances like autonomous driving scenes [106] because instances are then close to each other and not directly separable using the off-the-shelve features.

On the other hand, spatial clustering is the main force that drives unsupervised object discovery in 3D space [106]-[116]. In contrast to images, clusters are relatively easy to make in 3D space, but differentiating between clusters based on shape is hard because of the sparsity of the point clouds. Therefore, temporal tracking is often used to identify the dynamic clusters which are most likely objects such as walking pedestrians and driving vehicles. Self-training is the common process to also learn to detect static foreground objects. The intuition behind this is that a model trained on dynamic objects-only is not good at differentiating between static and dynamic objects with a similar shape. As a result, when such a model is used for inference, it will also detect a lot of static objects. The predicted objects are then used for re-training the detector,

and this is repeated multiple times until performance convergence. The drawback of this self-training is that the model gets a contradicting signal during training and training takes significantly longer due to the many rounds.

We argue that multi-modal data should be used jointly for unsupervised object detection as each modality has its own strengths, e.g. cameras capture rich semantic information, LiDAR provides accurate spatial information, and radars offer instant velocity estimation. Existing work [106] does use multi-modal data for unsupervised object detection but they do not use the modalities jointly. They split the training procedure into two parts: (1) training a detector using LiDAR-based pseudo bounding boxes, and (2) alternating between training a camera-only detector using the outputs of the LiDAR-only detector and vice versa. They note that the appearance of static and dynamic foreground objects is similar in camera images and that the camera-only model will not be able to distinguish them. However, their method ignores the fact that both modalities can be used at the same time for creating pseudo bounding boxes.

2.7.1 Multi-modal 3D object detection

We propose the method UNION (UNsupervised multI-modal 3D object detection) that exploits the strengths of camera and LiDAR jointly. We extract object proposal clusters from the LiDAR point cloud, using ground removal and spatial clustering, and employ temporal tracking to identify the dynamic objects from the proposals. Then, we leverage the camera images to encode the appearance of each object proposal. We exploit the appearance similarity between static and dynamic foreground instances. We cluster all object proposal representations, and differentiate between background and static foreground instances by selecting the static object proposals that have a similar appearance embedding as dynamic object proposals. Finally, the identified objects are used to generate pseudo bounding boxes and pseudo class labels which can be used to train any existing object detector in an unsupervised manner using their original training protocol. Figure 16 shows the overview of the UNION framework, in which a sequence of LiDAR and camera data is used to extract static and dynamic foreground object proposals from the scene. Dynamic foreground objects are obtained using spatial clustering and temporal tracking, and these are used to find the static foreground objects by clustering the appearance of each object proposal. For all discovered objects, a pseudo bounding box is generated and a pseudo class label is assigned based on the appearance embedding.

The task of 38ptimit detection is to detect objects in 3D space. Here, we consider the 2D BEV (bird's-eye view) detection representation [80] and each BEV bounding box b = (x; y; l; w; θ) consists of the center position (x; y), the length and width (l; w), and the heading θ. In the case of supervised learning, there are in general K unique classes c_k defined and each object instance has one of the class labels. This is called multi-class

object detection, and methods can learn the existing classes during training. However, for the unsupervised case, class labels are only available during evaluation. Hence unsupervised methods cannot learn the definition of the existing classes as with supervised learning and are for evaluation limited to class-agnostic evaluation.

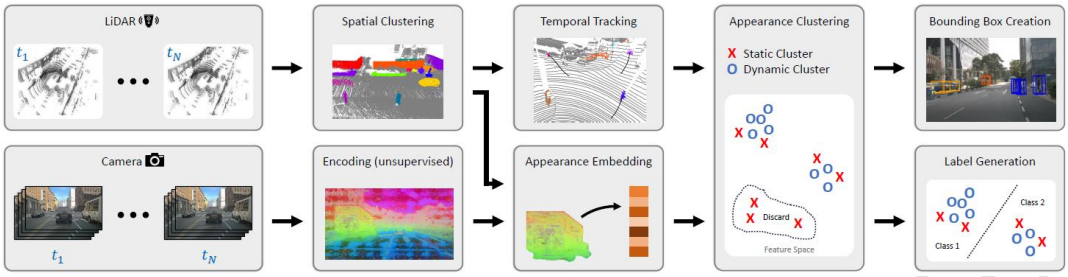


Figure 16: Overview of the UNION framework.

We want to discover objects in 3D space using LiDAR data, camera data, and multiple sequences of T frames. For each frame t, we assume that we have a LiDAR point cloud and camera images. We indicate the sequence data of LiDAR and camera with P and I, respectively. UNION uses an entire sequence to generate pseudo bounding boxes and pseudo class labels, and it consists of multiple blocks, see the overview above.

We want to create pseudo bounding boxes for both static and dynamic foreground instances because the supervision signal for training is then more consistent, e.g. both parked and driving vehicles are considered as positive examples. Besides, otherwise, the model may learn a bias where dynamic objects occur while this may be different for static objects. Lastly, both considering static and dynamic objects enlarges the training set for training the detector.

We exploit the accurate spatial information of the LiDAR for getting object proposals. First, we fit a plane in each point cloud to remove the ground plane points and all points below the ground. Then, we aggregate the non-ground points of k LiDAR scans into a single coordinate system, e.g. the coordinate system of the ego vehicle at the first-time frame of the k scans. After that, we use spatial clustering to divide the aggregated point clouds into M different segments. This means that all points that do not belong to the ground have a segment ID m. Subsequently, we use temporal tracking with a linear velocity model to determine which segments are dynamic, and which segments are static. This gives us for each time frame two sets, namely the set of dynamic segments and the set of static segments. Both sets contain object proposals. We are certain that the set with dynamic segments consists of foreground objects but the set of static segments contains both parts of the background (e.g. houses, poles, and bridges) and foreground objects that are currently not moving (e.g. standing pedestrians and parked vehicles).

We use the camera images to be able to differentiate between background and static foreground instances. We encode the camera images of each time frame t using an off-the-shelve vision foundation model [91] trained with self-supervised learning. The foundation model computes a feature map for each camera image. Bilinear interpolation is used to upsample the spatial resolution to the spatial size of the image such that each pixel in the original image has a corresponding feature in the upsampled feature map. Subsequently, we project for each segment the LiDAR points to the image plane and assign to each point a camera-based feature vector. After that, we append to each feature vector the 3D coordinate with respect to the segment centroid, i.e. we subtract the 3D coordinate of each point by the centroid of the segment. All obtained vectors for a single segment for a time frame are then used to calculate the appearance representation of that specific segment by averaging the feature vectors. After aggregating the vectors of a segment, we have for each sequence M segments and T time frames, thus in total T *M representation vectors.

We cluster the segment representations using k-nearest neighbours algorithm to find the static segments that have a similar appearance as the dynamic segments and the background segments that do not have a similar appearance as the dynamic segments. We cluster the representation of multiple sequences in the same space such that we have more diverse segments. By doing this, each static segment gets a label that indicates whether it is 'background' or a 'static foreground object'. Thus, we split the set of static segments into two disjoint sets, namely the set which is considered to have the static foreground objects and the set which is considered to have the background segments. After that, the representations of the background segments are removed from the representation space, and the remaining representations, i.e. the static and dynamic foreground objects, are clustered and to each cluster, a cluster ID is assigned. These can be considered to be pseudo class labels, and thus we can use those labels to train a multi-class detector. We use the LiDAR points of the segment to fit a 3D bounding box around it.

2.7.2 Datasets

We evaluate our method on the challenging nuScenes [66] and Waymo Open Dataset [96] datasets. Both datasets provide 3D point clouds and 2D RGB image data that are suitable for our task setting. It is of great significance to verify our unsupervised method under such a real and large-scale complex scene.

The nuScenes dataset is a large-scale autonomous-driving dataset for 3D detection and tracking, consisting of 700, 150, and 150 scenes for training, validation, and testing, respectively. A scene is a sequence of roughly 20 seconds, and the data is labelled with 2Hz. Each frame contains one point cloud and six calibrated camera images covering the 360-degree horizontal field of view (FOV). The dataset provides the transformations between all sensors and between time frames.

The Waymo Open dataset Is also a large-scale dataset for autonomous driving. We utilize point clouds from the 'top' LiDAR (64 channels, a maximum distance of 75 meters), and video frames (at a resolution of 1280x1920 pixels) from the 'front' camera. The training and validation sets contain around 158k and 40k frames, respectively. All training frames and validation frames are manually annotated with 2D bounding boxes and 3D bounding boxes, which are capable of evaluating the performance of 2D object detection and 3D instance segmentation.

2.7.3 Baselines

We use different baselines for nuScenes and Waymo. For nuScenes, we compare against the three unsupervised baselines, namely (1) HDBSCAN [89], (2) PP score [115], and (3) MODEST [115]. For WOD, we use the same baselines as for nuScenes. Besides, we compare to Wang et al. [106]. For all baselines, we train BEVFusion [88] to evaluate the performance of the method if the method is compatible with standard object detectors. If the method is not compatible with standard detectors, we use the predictions of their trained models on the training data to get the pseudo labels. Lastly, for both datasets, we also compare our method to training BEVFusion using supervised learning. We cannot compare to the methods of Chen et al. [70], Najibi et al. [90], and OYSTER [116] as those methods have not released their source code and did not use the same datasets and/or settings as our other baselines.

HDBSCAN performs density-based spatial clustering, grouping points with many nearby neighbours together. We use this mechanism to get clusters in the LiDAR point cloud, filter the clusters based on size, and fit a bounding box around each cluster. PP score estimates the persistence of a point by measuring the variance in the number of LiDAR points within the point's neighbourhood encountered in other observations of the same region. These PP scores can then be used as a feature for clustering and retrieving mobile objects that have a low PP score. Note that in the original paper, multiple traversals over the same area are assumed, which poses a very strict requirement for data collection. To avoid imposing this strict requirement, we only consider a single traversal with multiple observations over a short period of time, i.e. the length of the sequence. MODEST (1 traversal) adds two rounds of self-training after the initial training compared to PP scores, where at each round the pseudo labels coming from the latest self-trained model are filtered using PP scores to discard persistent clusters.

2.7.4 Performance metrics

For nuScenes, the main metric that we consider is the Average Precision (AP) [74]. The AP is defined by the BEV center distance instead of the 3D intersection over union (IoU), and we use the standard distance thresholds for nuScenes, namely 0.5m, 1m, 2m, and 4m. Evaluation is conducted on the annotated validation split and we use the

entire horizontal field of view for evaluation. We do a class agnostic evaluation, i.e. we only consider the foreground class. So, we do not calculate the mean Average Precision (mAP).

For Waymo Open dataset, we follow the evaluation protocol of Wang et al. [106] that evaluates the detections as 3D instance segmentations. Evaluation is conducted on the annotated validation set of the Waymo Open dataset. We evaluate the performance of 3D object detection. The dataset contains four annotated object categories, namely 'vehicles', 'pedestrians', 'cyclists', and 'sign'. We test the class-agnostic average precision (AP) score for vehicles, pedestrians, and cyclists. Wang et al. note that for 3D instance segmentation, no previous metrics have been proposed on WOD. They propose to compute the 3D AP score based on the IoU between predicted instance point sets and the ground truth. The ground truth for the instance segmentation can be obtained by labelling the point within 3D bounding boxes. The 3D AP score is reported at the point sets IoU threshold of 0.7 and 0.9, denoted as AP70 and AP90, respectively.

2.7.5 Preliminary results

The spatial clustering component first segments the LiDAR point clouds into ground and non-ground points. After that, the non-ground points are clustered, and a 3D bounding box is fitted for each cluster. Figure 17 shows a segmented point cloud for an example scene. It can be seen that the round and sideways are correctly segmented as ground and that objects like parked vehicles (bottom) are labelled as non-ground points.

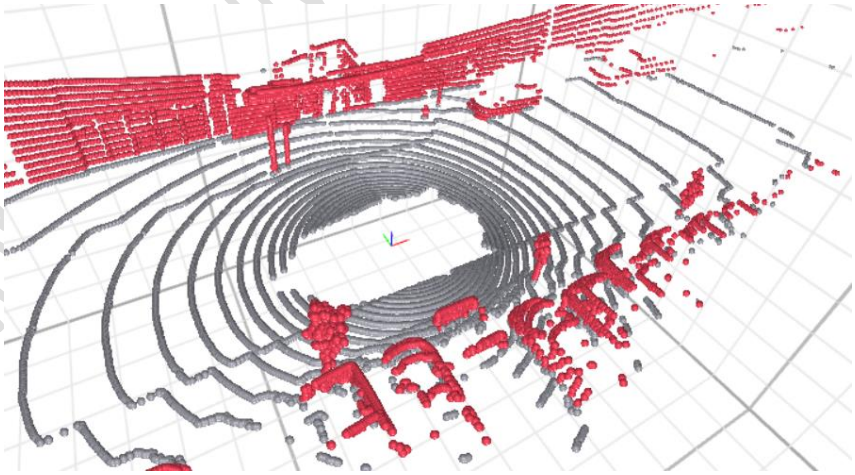


Figure 17: A LiDAR point cloud segmented into ground and non-ground points.

Figure 18 shows the clustered non-ground points together with the fitted bounding boxes. It can be seen that each parked vehicle has its own cluster and bounding box. The bounding box fitting algorithm optimizes an objective that aims at selecting a bounding box where most points are close to the sides of the bounding box. The large

wall that is visible at the top of the figure also is a cluster with its own bounding box. These can easily be filtered by removing clusters of which the bounding box is larger than a certain threshold.

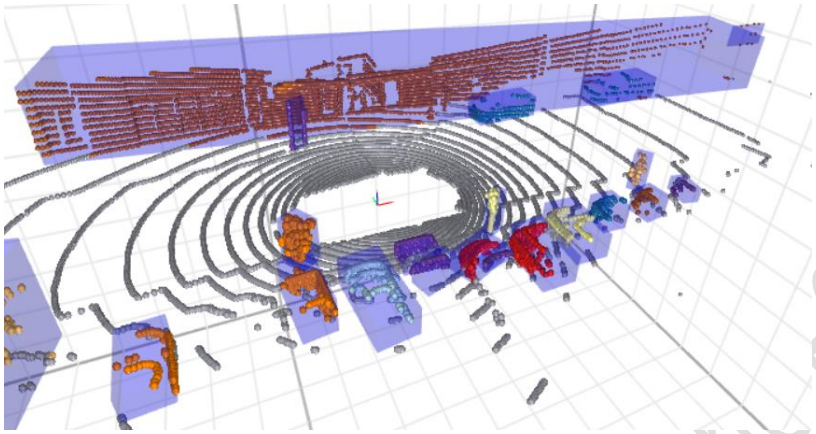


Figure 18: The spatial clusters together with their fitted bounding boxes. Note: The ground points are shown in gray

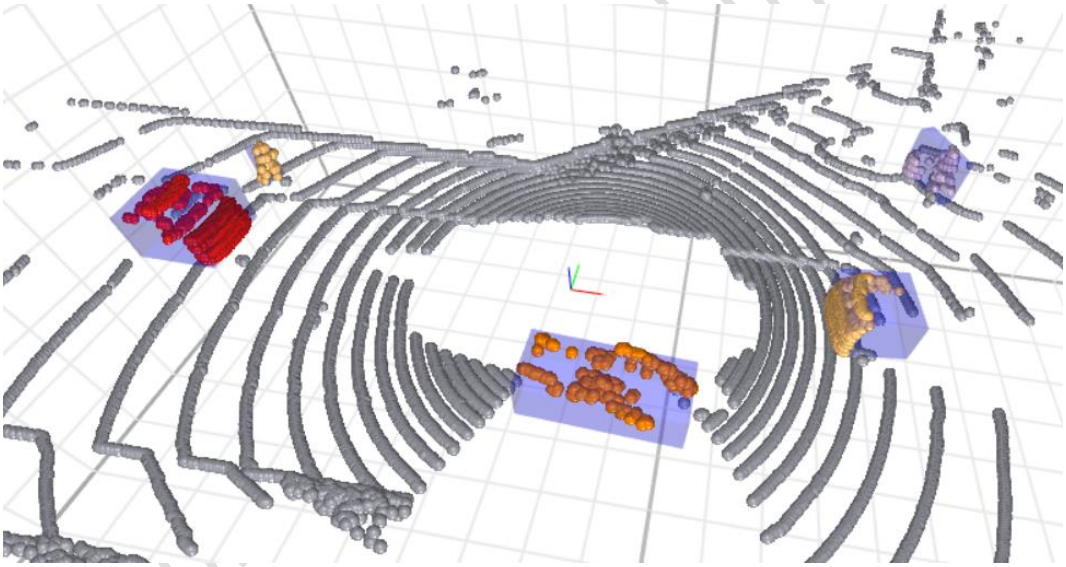


Figure 19: The spatial clusters of a single frame that have a velocity larger than 0.10 m/s.

The temporal tracking component uses scene flow estimation to determine a velocity for each spatial cluster. Figure 19 shows the clusters that have a velocity larger than 0.10 m/s. It can be seen that there are 5 moving clusters of which 3 are driving vehicles and 2 are walking pedestrians. The clusters with a velocity lower than 0.10 m/s are not shown. These clusters are considered to be static.

The encoding component computes for each camera image a feature map and the appearance embedding component uses the feature maps to determine for each spatial cluster a camera-based feature vector. Figure 20 shows the distribution of the first 2 PCA components for the ground truth bounding boxes of the nuScenes dataset together with the ground truth class labels. It can be seen that the classes that can

move such as car and pedestrians are on the left side of the figure, while classes that cannot move by itself such as barriers and traffic cones are on the right side of the figure. This supports the hypothesis that instances of the same class have a similar appearance embedding and that moving behavior of appearance clusters can be used to differentiate between types of objects.

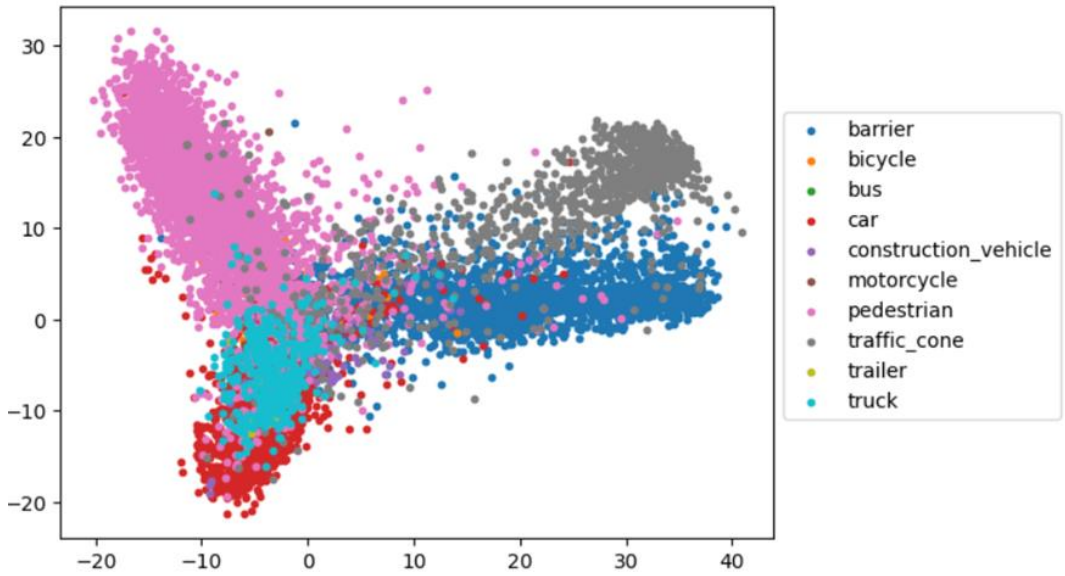


Figure 20: The distribution of the first 2 PCA components for the ground truth bounding boxes of the nuScenes dataset together with the ground truth class labels.

2.7.6 Ongoing & future work

The appearance clustering component is still work in progress. For each cluster, there is a camera-based appearance embedding and a velocity that can be converted to a Boolean indicating whether the cluster moves by using a velocity threshold. The next step for the appearance clustering component is to cluster the appearance space and use the velocity information to select the clusters containing moving instances. After that, the clusters that remain can be used to train existing 3D object detectors on the nuScenes dataset.

The future work is listed below:

- Finish appearance embedding component, and use pseudo bounding boxes and pseudo class labels to train existing 3D object detectors.
- Experiment with more datasets including the View-of-Delft and Eurocity Persons v2 datasets.
- Prepare submission to ECCV 2024.

  Collect new data with our vehicle, train existing detectors using our method, and test the trained detector on the fly.

3. Semantic scene analysis and precise45 localisation

3.1 Introduction

The aim of this task is to obtain an instantaneous spatial and semantic representation of the environment, related to both static (i.e. infrastructure) and potentially dynamic elements (i.e. road users) using on-board sensing. Relevant spatial representations include point clouds, 3D bounding boxes, 2D/3D occupancy grids, pillars, or implicit shape representations. This task also involves the identification and automatic extraction of object attributes ("context cues") which are indicative of road user intent and can be used for motion /behaviour modelling and path prediction (cf. Task3.3) e.g. pose, clothing (e.g. uniform of police officer), together with elements of the static environment (e.g. road lay-out, road markings, traffic lights), and elements of the dynamic environment (other road users). The techniques in this task will predominantly involve deep learning and will be focused on scene completion against missing scene elements due to occlusions.

In addition, robust and accurate localisation will be developed based on on-board LiDAR sensors (working continuously even in absence of HD maps or GNSS-denied environments). The focus here is to filter out environmental outliers such as snow, rain and fog and allow autonomous vehicles to drive on roads without lane markers and landmarks. For this purpose, an approximate 3D map of the environment based on 3D LiDAR data represented as "volumetric probabilistic distributions" will be generated as part of a simultaneous localisation and 3D mapping component. Global positioning can be obtained on top of LiDAR-based positioning by fusing together GNSS (using the Arctic Galileo stations) and ego-vehicle inertial data.

3.2 EXP4: Roadworks, unmarked lanes and narrow roads

3.2.1 Background/Problem statement

Semantic Scene Analysis

EXP4 requires a 2D object detector that can both classify base classes (pedestrians, vehicles, and cyclists) as well as signs that would inform the autonomous vehicle that it may encounter road works. EXP4 seeks to retrain an off the shelf open-source object detector to detect both the base classes and traffic signs that provide semantic information related to road works.

Precise Localisation

Continuous and highly accurate vehicle pose is critical for EXP4. In particular, given the absence of any lane markings to denote lane boundaries the autonomous vehicle

would need to rely on high-definition maps and accurate localisation in order to operate safely. To this end, we address the need to develop accurate and robust localisation in as many scenarios as possible (including when GNSS is denied). The work carried out will focus on the following:

1) Investigate state of the art LiDAR based SLAM methods for aiding in localisation when GNSS is degraded.
2) Investigate how we can improve the accuracy of the constructed maps through several mapping runs.

3.2.2 Approach

Semantic Scene Analysis

The 2D object detector used in this experiment is Yolov5 [129]. We compiled the following datasets for training and testing our object detector, in particular:

1. Yolov5 (pre-trained model): this is Yolo's own pre-trained model which serves as a starting point to set a base line on detections.
2. Yolov5 trained v1: The first version is trained just on the COCO dataset with the patched augmented signs for 1000 epochs with the default Yolo hyper-parameters (see Figure 21a).
3. Yolov5 trained v2: The second version is trained on the combined COCO dataset with the patched augmented signs along with the NuImages dataset. Training was for 700 epochs with the default Yolo hyper-parameters (see Figure 21b).
4. Yolov5 trained v3: The third version contains additional processing on the COCO dataset. In addition to the original images with patch augmentations, COCO now contains further patched traffic signs on the original images plus their flipped counterparts. This greatly increases scene variety. The flipped COCO dataset is combined with nuImages and trained for 700 epochs with the default Yolo hyper-parameters (see Figure 21l).
5. Yolov5 trained v4: The fourth and final version extends version 3 by including additional background (objects with no annotation and bounding box) traffic signs different to the ones being detected to improve robustness. Model is trained again for 700 epochs with the default Yolo hyper-parameters (see Figure 21d).

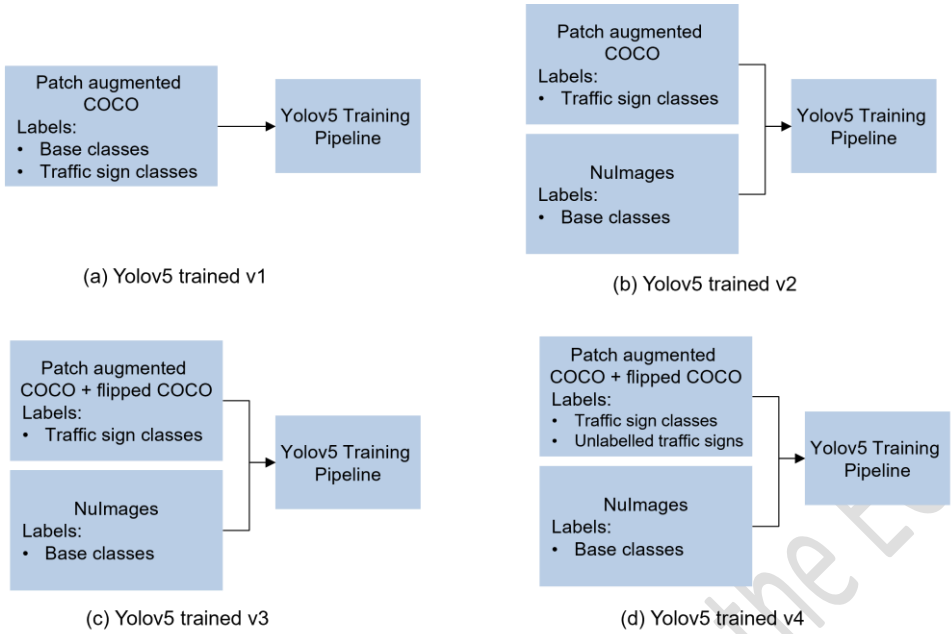


Figure 21: Diagrams showing datasets and corresponding classes for the different retraining runs.

The trained models are evaluated both quantitatively and qualitatively. For quantitative evaluation the mean average precision0.5:0.05:0.95 (mAP0.5:0.05:0.95) metric is used [130]. The mAP0.5:0.05:0.95 is derived by calculating the precision and recall. Precision is the ratio of true positive predictions to the total number of positive predictions (true positives + false positives). It measures the accuracy of positive predictions. Recall is the ratio of true positive predictions to the total number of actual positives (true positives + false negatives). It measures the ability to find all positive instances. We can compute both precision and recall over several intersection over union (IoU) thresholds in order to obtain a precision-recall curve. In turn the average precision (AP) is calculated for each class in the dataset. It represents the area under the precision-recall curve. It gives a single-value measure of the classi'ier's ability to discriminate between positive and negative examples. The mAP0.5:0.05:0.95 is then simply calculated by taking the average of the AP values across all classes. This provides an overall measure of the m'del's performance across different object categories.

Table 4: Base class performance (nuImages). mAP0.5:0.05:0.95

Class Model	Pedestrian	Bicycle	Car	Motorcycle	Bus	Truck
Yolov5 pre-trained	*0.40*	*0.30*	*0.53*	*0.42*	*0.44*	*0.33*
Yolov5 Trained v1	*0.34*	*0.15*	*0.43*	*0.33*	*0.47*	*0.28*
Yolov5 trained v2	*0.27*	*0.10*	*0.39*	*0.27*	*0.34*	*0.23*
Yolov5 trained v3	*0.36*	*0.46*	*0.55*	*0.44*	*0.49*	*0.45*
Yolov5 trained v4	*0.39*	*0.52*	*0.56*	*0.46*	*0.55*	*0.44*

Table 5: Traffic sign class performance (Mapillary). mAP0.5:0.05:0.95

Class Model	Speed Limit 20	Speed Limit 30	Speed Limit 40	Speed Limit 50	Speed Limit 60	Roadworks
Yolov5 pre-trained	*0*	*0*	*0*	*0*	*0*	*0*
Yolov5 Trained v1	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Yolov5 trained v2	*0.15*	*0.22*	*0.32*	*0.27*	*0.07*	*0.11*
Yolov5 trained v3	*0.13*	*0.25*	*0.36*	*0.24*	*0.15*	*0.08*
Yolov5 trained v4	*0.25*	*0.30*	*0.45*	*0.42*	*0.15*	*0.19*

The final model to be used for EXP4 and EXP5 is the "Yolov5 trained v4", as based on the mAP scores in Table 4 and
Table 5, we are able to detect both the base classes and corresponding traffic signs of interest with one single model.

Precise Localisation

When the ego-vehicle enters an area with degraded GNSS signal and without lane markings, a LiDAR can provide a precise localisation alternative due to its high accuracy in depth measurements compared to vision sensors. However, most LiDAR based localisation work faced the challenge of processing large number of points returned from LiDAR for point cloud registration [129][132]. This challenge makes it difficult for having a real-time and precise LiDAR based localisation.

Recently, a Direct LiDAR Odometry (DLO) algorithm that is computational efficiency, consistent, robust, and accurate performance in real world dataset was introduced [133]. In order to realize high-speed and high accuracy, DLO algorithm implements an adaptive keyframing system, a keyframe-based sub-mapping approach and a

lightweight GICP (Generalized Iterative Closest Point) [134] solver called NanoGICP. Firstly, the pre-processing is performed to reduce noise or redundant LiDAR points. In addition, the point cloud can down sample by a voxel grid filter to reduce processing time. Then the scan-to-scan matching is performed by GICP to estimate the pose change between previous scan and current scan. Next, the scan-to-map matching is performed to estimate final pose in the 3D sensor map. Simultaneously, the 3D sensor map is updated by map matching result. We adapted and tested the DLO algorithm on both public dataset (KITTI dataset) and our custom dataset with known initial pose and taking only the input from point cloud. The preliminary results show the accuracy of DLO algorithm in both localisation and mapping. Figure 22 shows a point cloud map that is built from DLO algorithm. Figure 23 presents the odometry generated from GPS data and DLO output overlayed on a map. In an area where the GPS signal was weakened, we can see that the localisation output from the LiDAR based algorithm can maintain a good accuracy.

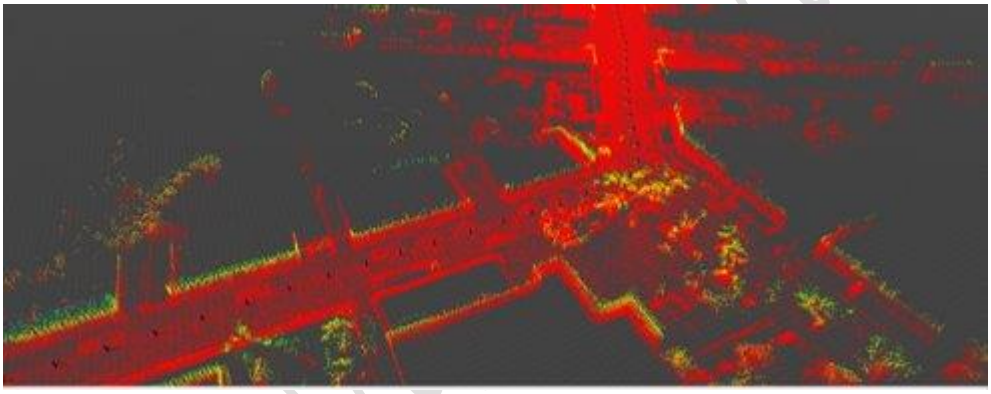


Figure 22: A map generated from LiDAR point cloud using DLO.

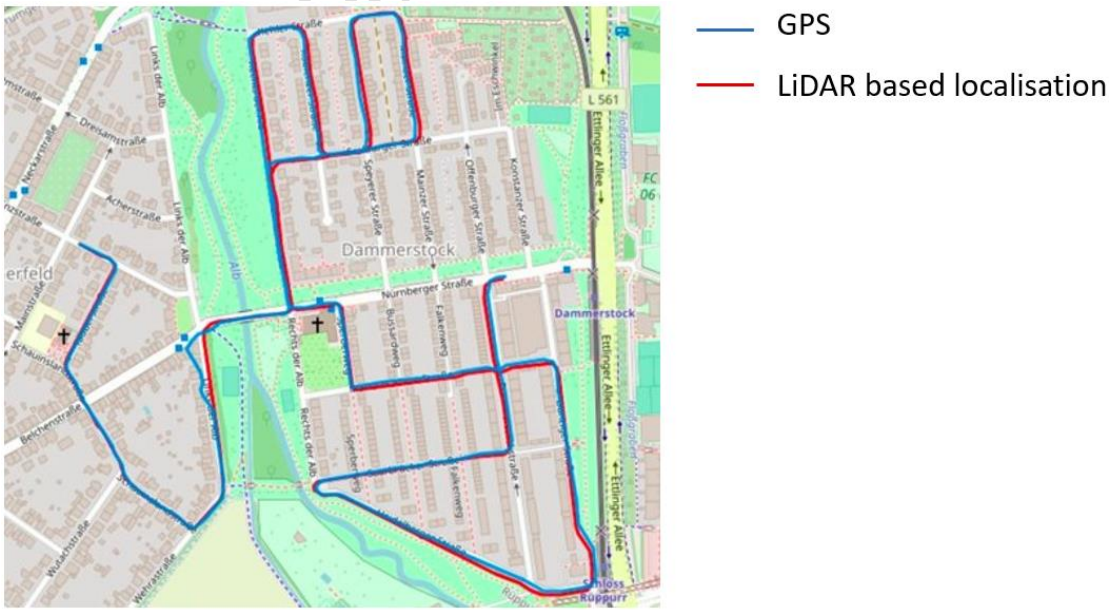


Figure 23: Comparison of the generated trajectory from GPS data and from DLO algorithm on KITTI dataset.

Over a long trajectory, the LiDAR based odometry can be drifted from the ground truth location. Therefore, the estimated keyframe pose from DLO can be fused with IMU, wheel odometry, and GPS to increase the accuracy of the final estimated pose as depicted in Figure 24.

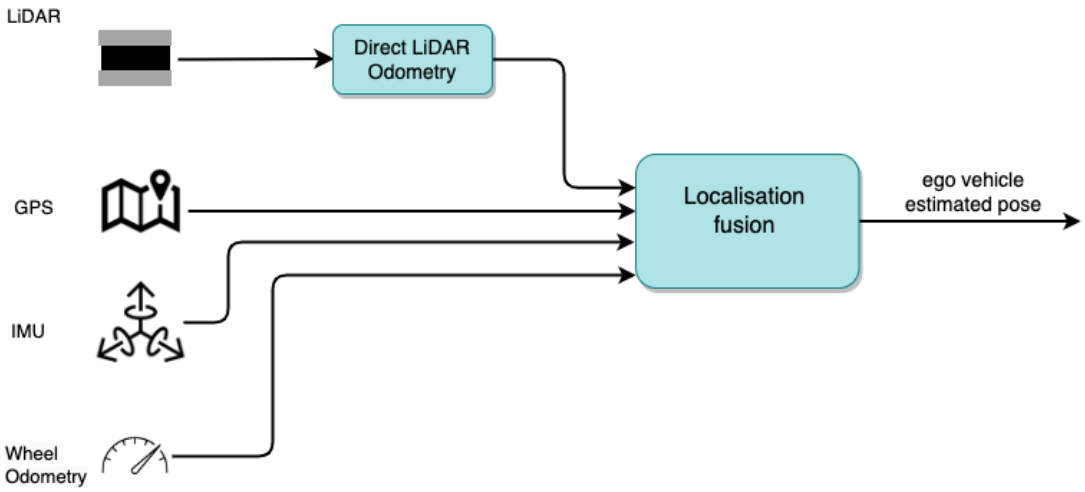


Figure 24: Diagram of the localisation fusion.

The second approach relies on improving the accuracy of SLAM maps using several mapping runs. In this approach we consider the following scenario (see Figure 25 for block diagram):

- The SLAM (or other approaches) method is first used to map an area.
- The stored map can be used for future vehicle relocalisation.

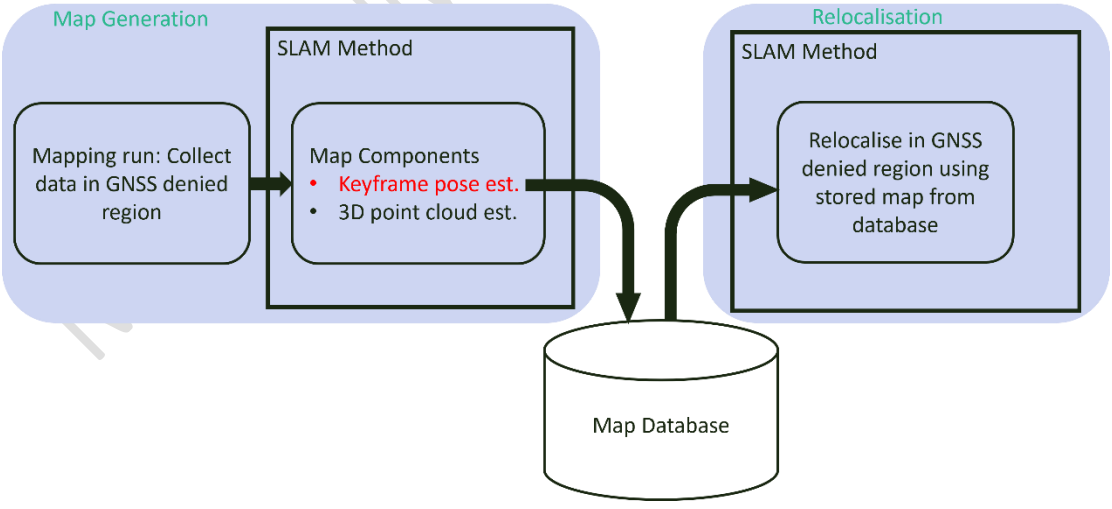


Figure 25: Block diagram showing the general paradigm of mapping a GNSS denied route using a SLAM like method. The mapped route can be used for future relocalisation.

Constructing a map for relocalisation requires accurate estimation of keyframe pose. To this end, we propose to formulate a joint optimisation framework for more accurate estimation LiDAR keyframe pose. A block diagram for the potential approach

is shown in Figure 26. Our approach is similar to the work in [135], where multiple robots were used to generate a single map of an area via the formulation of a graph optimisation problem where temporal constraints for each individual robot were enriched with spatial constraints between the different robots. Our proposed approach seeks to utilise the spatial constraints in order to improve the maps built along a specific route. Furthermore, the approach being considered some form of loop closure is possible and accurate initial pose is available.

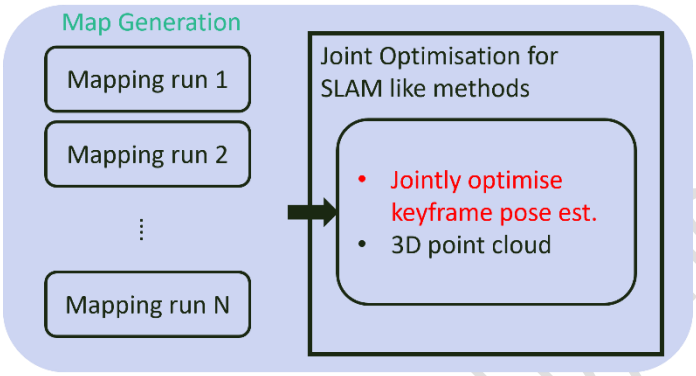


Figure 26: Proposed approach to jointly optimise several mapping runs in order to estimate keyframe pose more accurately.

Evaluation will be considered both on synthetic and/or real-world data (RTK-GPS would be considered ground truth for real-world data). Metric such as root mean square error (RMSE) would be used for quantitative evaluation.

3.3 EXP6: Far range small object detection in adverse weather

This section was contributed by project partner APTIV.

3.3.1 Background/Problem statement

A radar-based solution was chosen as it offers advantages over other sensing modalities when the visibility is impaired (night-time, low sun, heavy rain, snow and fog) which are the situations in which the user would require the most help from an ADAS solution and where the user would still expect good performance from a fully autonomous system.

Perceiving such obstacles is particularly challenging for a radar-only solution due to the numerous physical effects involved in the scenario such as strong reflectors like guardrails and overtaking vehicles. There are a few studies that address debris detections using radar [137][138][139][140] but many of them either consider a very limited set [137][138][139] or limited range [137][138][140] and do not estimate the debris height, showing that further studies are required.

3.3.2 Approach

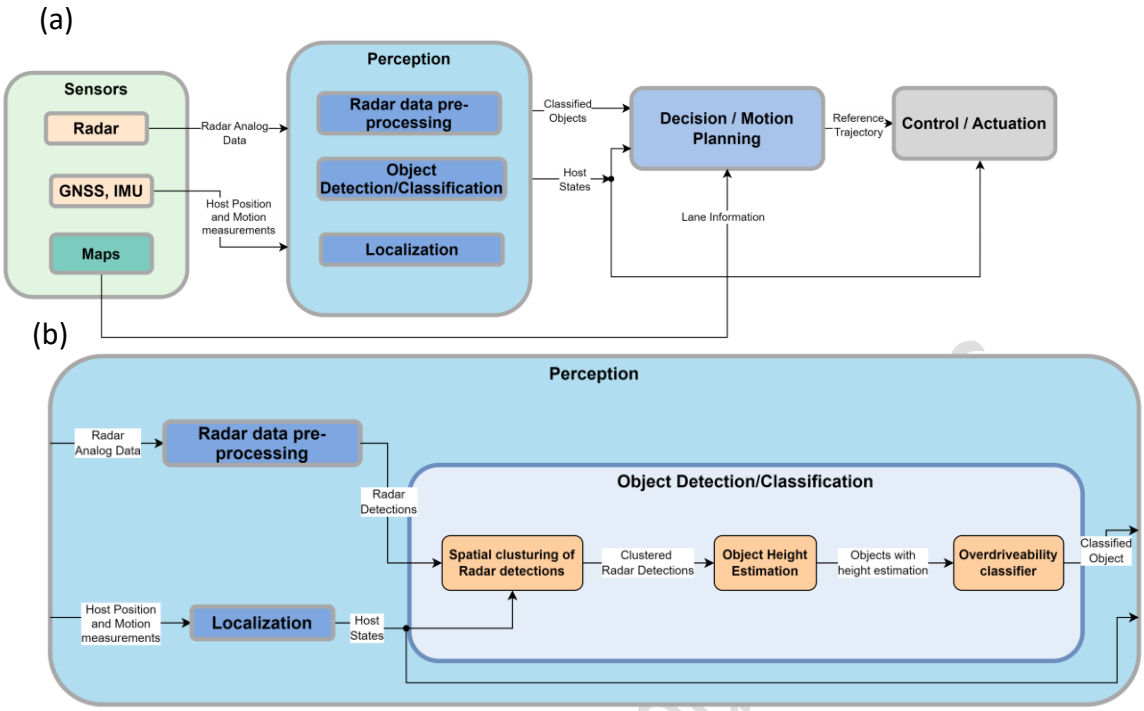


Figure 27: EXP 6 Architecture. (a) High-level Architecture and interfaces. (b) Detailed Full Stack Architecture and Interfaces from [157].

This section first describes the methods required to implement the "Spatial Clustering of radar detections" module shown in Figure 27 within the Perception component. (Subfigure "a" provides the context of the Perception and subfigure "b" the module decomposition within Perception). The proposed perception method requires two inputs: radar detections and host states. The radar detections are provided by the "Radar data pre-processing" module and the 6D host states are provided by the "Localization" module shown in Figure 27 b. The radar detections are assigned a location in 3D space in the vehicle coordinate system and contain several attributes such as range (distance), range rate, azimuth and elevation angles. The latter is critical for Experiment 6 as without it height cannot be estimated accurately. This is possible as we are using a 4D imaging radar [141]. The most important host states for the perception module are the velocity, yaw rate and pitch of the ego vehicle. The pitch is required as it can influence height estimations.

The aim of the "Spatial Clustering of radar detections" is to group or cluster the radar detections in a region of interest (the region in front of the vehicle along a straight lane, refer to 0 for further details) to enable height estimation for individual objects. Data points can be clustered together using algorithms like K-means and DBSCAN [142].

K-means is a partition-based clustering algorithm [143] that iteratively segments the data in k clusters around the nearest available k mean values; the number of clusters needs to be predefined. The algorithm then recomputes the new k mean values and continues until there are no changes in the cluster's computed mean values [144]. One major drawback of the K-means algorithm is that it produces convex-shaped clusters and does not fit well to arbitrarily shaped objects [144][145]. Additionally, partition-based algorithms are sensitive to outliers [143].

DBSCAN (Density Based Spatial Clustering of Applications with Noise) [145] is a density-based clustering algorithm which means that data in regions where the density is higher than its surroundings is grouped as one cluster. DBSCAN requires the user to define two parameters: the radius (Eps) around each point considered as its neighbourhood and the number of points (MinPts) required to form a cluster. The algorithm starts with a random point and iterates through each point looking for core points; clusters are formed around core points. A core point is a point which has at least MinPts points in the neighbourhood radius Eps. Different clusters will be separated by a distance of at least Eps. DBSCAN is the most well-known density-based clustering algorithm [146], and there are many proposed solutions [147][148][149] that use this algorithm to cluster radar detection data.

We now turn to describing the methods required to implement the "Object Height Estimator" and "Overdriveability classifier" modules shown in Figure 27(b). The clustered radar detections accumulated over time are used as the input of the "Object Height Estimator" module. The output of this module is then used as input for a simple classifier.

Owing to the lack of publicly available radar datasets that provide the elevation angle, not much literature could be found about height estimation from 4D radar sensor data. Paek et al. [200] compute 3D bounding boxes for large objects (pedestrian, motorcycle, bicycle, car and truck/bus) in different weather conditions using the power measurements along the Doppler, range, azimuth and elevation dimensions as input to a Neural Network. The Neural Network consists of a pre-processing block, a 3D Sparse convolution backbone and a final bounding box prediction module that uses an anchor-based method. A previous study from APTIV by Tyagi et al. [201] also uses 4D radar data as input but does not use the elevation component in the study which is about early debris detection; the height of the object is not estimated in this work but it focuses solely on object detection. They propose a solution that performs pre-processing on the range-azimuth maps and extracts features from it which are used as input to an LSTM network which outputs in-lane stationary detections; the output of the neural network is smoothed using a post-processing step that provides pseudo-probabilities of the detections.

Looking at the literature for 3D radar sensor data we can find parallels to our proposed approach (see Figure 27). Scheiner et al. [202] provide a comparison of five methods

to perform object detection: (1a) a two-stage clustering (velocity filter + customised DBSCAN) and an ensemble of recurrent neural network classifiers using LSTM, (1b) a two-stage clustering (velocity filter + customised DBSCAN) and a random forest, (2) Point-Net++ architecture to classify every detection and a velocity and background filter with customised DBSCAN clustering and a voting scheme for the class of each cluster, (3) grid mapping (one for the amplitude and two for the x and y components of the Doppler) and YOLOv3 method, (4) an improved PointPillars method which outputs classified bounding boxes directly from the radar point cloud input and (5) combination of the first two methods. A similar approach to the methods (1 a and b) just mentioned was also used by Schumann et al. [197] to perform object classification. Our proposed method is similar to methods (1 a and b) in [202] and the method in [197] where a machine learning model is applied to the radar detection data inside of each cluster. The main difference is that our proposed model is a regression model which outputs the estimated height of each cluster.

As part of the requirements and metrics for the experiments in this project (for further details refer to Deliverable D2.1 [156]), it is required that our model outputs a confidence estimate linked to each prediction. One possible way of doing this is by using intrinsic uncertainty quantification. Intrinsic uncertainty quantification is where the model outputs both a prediction and a level of uncertainty. Examples of such models are Gaussian processes, Bayesian neural networks and deep ensembles [203]. Ensemble models can be used to compute a mean and a variance for each required prediction by considering the predicted output from a collection of models that are either different model types or have different parameters [204]. A Gaussian Process provides a posterior mean and a predictive variance for every test input thus inherently providing a measure of confidence [205].

Finally, once a height value is assigned to each cluster, a prediction on whether an object is overdriveable or not is computed based on the estimated height. The height threshold recommended through our literature review is **12 cm**. Using the normal distribution outputted by our height estimation model, a value for confidence can also be computed by using simple statistical techniques such as z-scores tables [206].

3.4 EXP8: Driving on secondary roads under adverse weather

3.4.1 Background/Problem statement

Within the adverse weather focus of EVENTS, Perciv.AI, in collaboration with TU Delft, is targeting scenarios with heavy raining and wet surfaces. In these circumstances, 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 radars, although it is widely acknowledged in the industry that the last one is

influenced the least. Therefore, radar is an ideal candidate for perception in adverse weather.

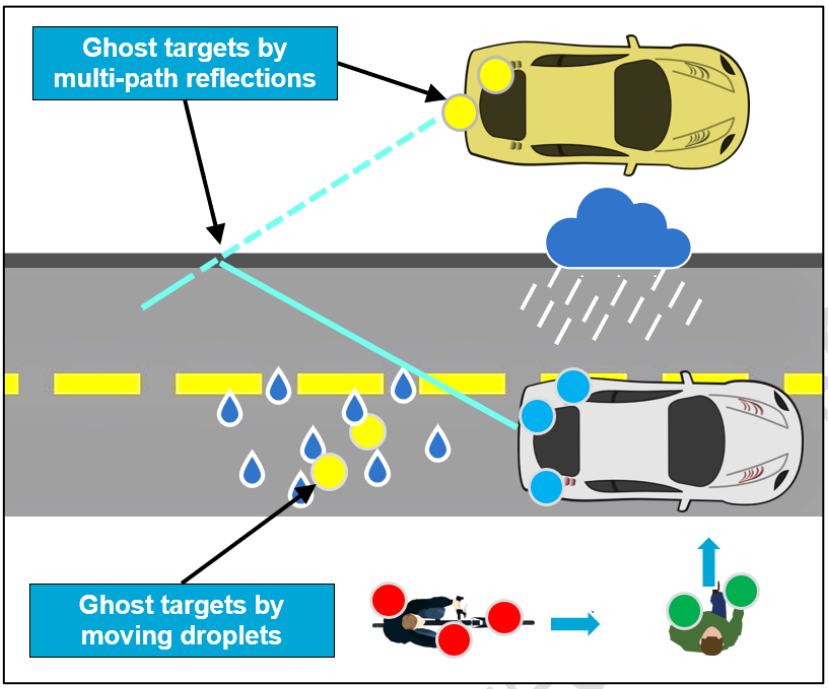


Figure 28 Illustration of radar point cloud's sparsity and noisiness.

However, solutions both in the literature and in the industry so far fall short to fully exploit the benefits of radars for such purposes given the following reasons:

1. Radars are relatively sparse and noisy sensors, especially compared to LiDAR sensors. Derived from their frequency domain, resolving close targets is difficult (sparsity) while its multipath propagation property introduces so called ghost targets, i.e. radar points that do not originate from their reported location. It is crucial that a filtering step is properly implemented to remove such clutter from the radar point cloud while keeping as many true points as possible, as the cloud is sparse as it is (see Figure 28).
2. Radars can measure relative radial velocities of detected points. This is a useful property for object classification [150], as velocity distribution can be highly class specific. In fact, in some approaches, dynamic and static objects are handled differently for this reason [151][152]. The disadvantage of these methods is that they depend on external odometry information, which can be compromised by the bad weather. Thus, it is important to have a solution which can estimate the motion of the ego-vehicle purely based on radar.

3. While radar-based road user detection methods are getting more popular in the last few years, these are either based on conventional approaches such as clustering [153] or on deep learning-based solutions designed for other sensors, usually LiDARs [154]. Proper, dedicated radar focused neural network development is missing in both industry and literature.

3.4.2 Approach

To address these three challenges, Perciv.AI is developing a multi-purpose, radar point cloud segmentation model as part of its EVENTS activities. The network will segment the input point cloud in three ways, addressing the three points above:

1. **Noise reduction:** each radar point will be assigned with a probability describing the likelihood of that point being "true" or a "ghost target". This output will be used for subsequent steps, but also in the self-assessment task T3.5.

2. **Static point detection:** each radar point will be assigned with a probability describing the likelihood of that point being a static or a dynamic target. This output will be used in the subsequent, purely radar based ego-motion estimation.

3. **Target classification:** After removing the ghost targets and distinguishing dynamic and static ones, the network will also assign a class probability for each radar target, including the classes of vehicle, cyclist, pedestrian, and background.

Combined, the network addresses all challenges listed before and provides valuable input for subsequent modules, such as ego-motion estimation [155], object detection [154], and the self-assessment task in T3.5.

Since Perciv.AI joined the project only recently, only preliminary results are available. E.g., the noise segmentation module is under active development with promising results, see Figure 29 below. Presenting both the radar input and the segmented occupancy grid (yellow – occupied, blue – free, green – unknown, red lines -sampling beams) both in top view, it is clearly visible how our network was able to clear up multiple noisy regions of the input, marked with red circles.

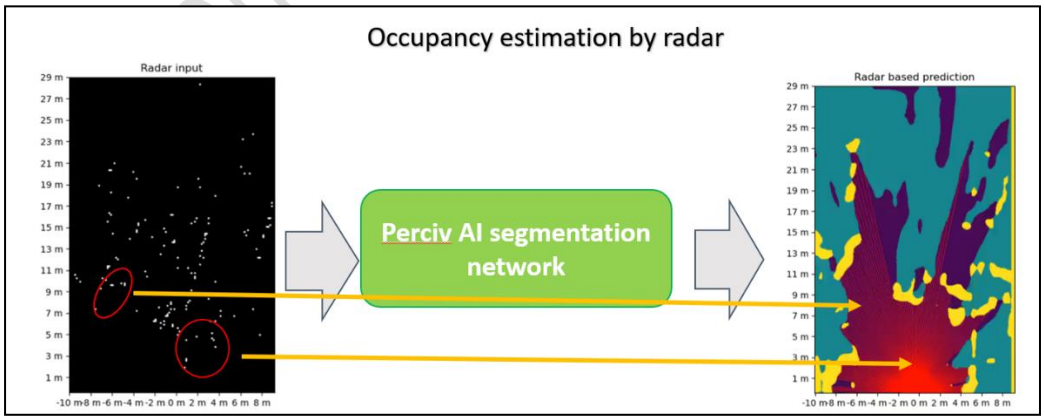


Figure 29: Noise segmentation in radar point clouds

Similarly, progress has been made on the static-dynamic segmentation challenges. On Figure 30 we show our latest experiment, demonstrating smooth and reliable ego-

motion estimation based on filtered radar point clouds. While there is room for improvement, the performance of the module was satisfactory for some debris detection challenges already.

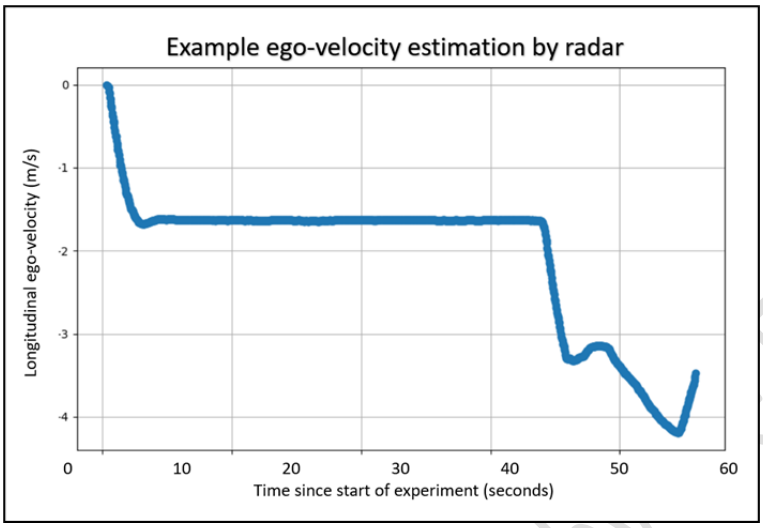


Figure 30: Ego-motion estimation based purely on radar data.

4. Environment state estimation and motion prediction

4.1 Introduction

This Section describes work on the integration of past and current measurements from on-board sensors to obtain the current environment state (incl. that of all relevant road users). Furthermore, it involves a prediction how the environment state will evolve over time.

4.2 EXP2: Re-establish platoon formation after splitting due to roundabout

This section describes work by project partner TECN.

4.2.1 Scope

The scope of this task underwent a shift within the perception efforts. It transitioned from its initial focus on predicting the motion of Vulnerable Road Users (VRUs) to the prediction of vehicle movements and the behaviour of dynamic obstacles within the context of EXP2. This initiative places a specific emphasis on addressing challenges related to occlusions within a roundabout and enhancing overall environmental perception in the domain of automated driving.

Within the framework of the EXP2 [212], a crucial component involves the incorporation of a collective perception module, intended to facilitate coordinated vehicle movements through a roundabout. The effectiveness of this module is dependent on the quality of the perception data provided by automated systems. In the context of this specific task, our primary objective was to enhance the detection and comprehension of CAVs, thereby improving our ability to predict their movements accurately. Below a simplified architecture focusing on CCAV perception module is shown, for a complete description of the modules, refer to D2.2 [213].

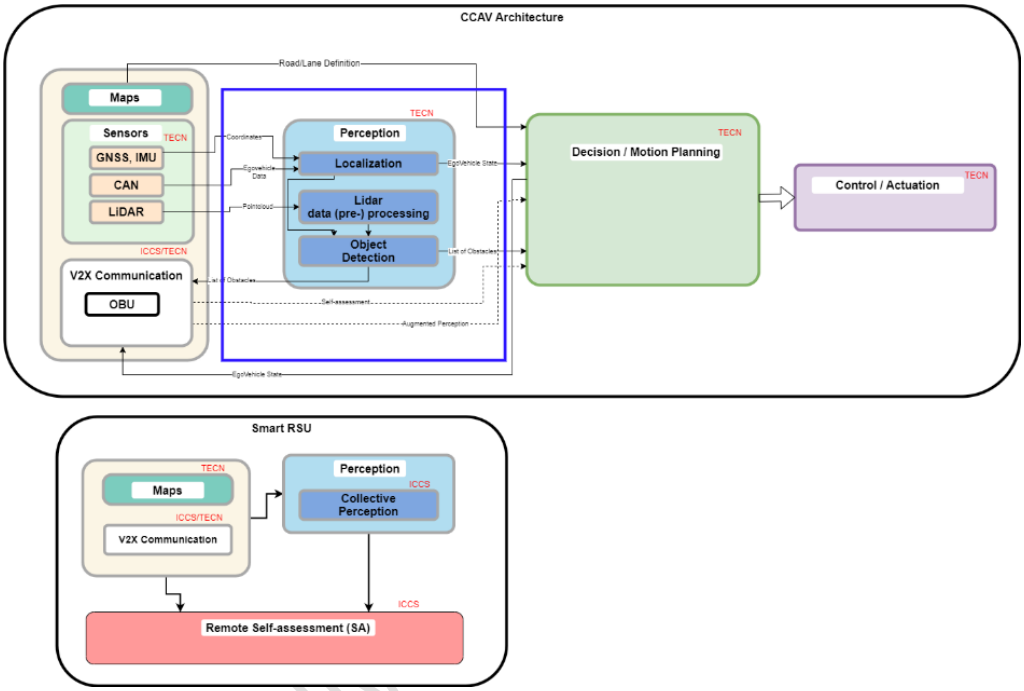


Figure 31: Simplified Architecture for EXP2 focused on Perception for CCAV.

4.2.2 Classification of motion prediction methods

There are several approaches to predicting the motion of the surrounding agents in a road scene. They are generally classified according to contextual factors that must be considered (physics-related, interaction-related and road-related factors) [214].

1. Physics-related factors refer to the kinematic and dynamic variables of the agents. They consider the spatio-temporal variables of the agents. Some examples of these variables are position, velocity, acceleration, etc.)

2. Interaction-related factors include the interdependencies and social rules between agents' manoeuvres. It is important to consider that traffic agents respect other agents and calculate their future paths taking this relationship into account.

3. Road-related factors include the relevant road regulations (lane type, traffic lights, stops, etc.) and the modelling of the map (usually HD map), including its topological, semantic and geometric information.

4.2.3 Motion prediction

We have adapted and trained HiVT [222] (Hierarchical Vector Transformer) to a map-free model that employs social interaction to compute multimodal predictions. The model calculates interactions that occur locally and globally between agents. The local encoder relies on a Temporal Transformer to predict multimodal predictions. Figure 32 shows the architecture proposal, where the modules are represented from the input to the output. The framework consists of three stages. The first stage encodes rotation-invariant local context surrounding each agent to make predictions. In this phase, ego-motion and neighbouring agents' motions are aggregated to provide valuable information about the scene. The second stage is responsible for the global interaction between agents. It considers the local context of different agents to update log-range dependencies and scene dynamics. Technical abbreviations such as 'log-range dependencies' should be explained when first used. Finally, the representations are utilised to produce multimodal forecasts for all the agents.

The 20 previous frames of the agents in the scene comprise the input model. The model output consists of six predictions per agent, each worth 30 points. Each trajectory carries its own confidence to measure the probability of that trajectory.

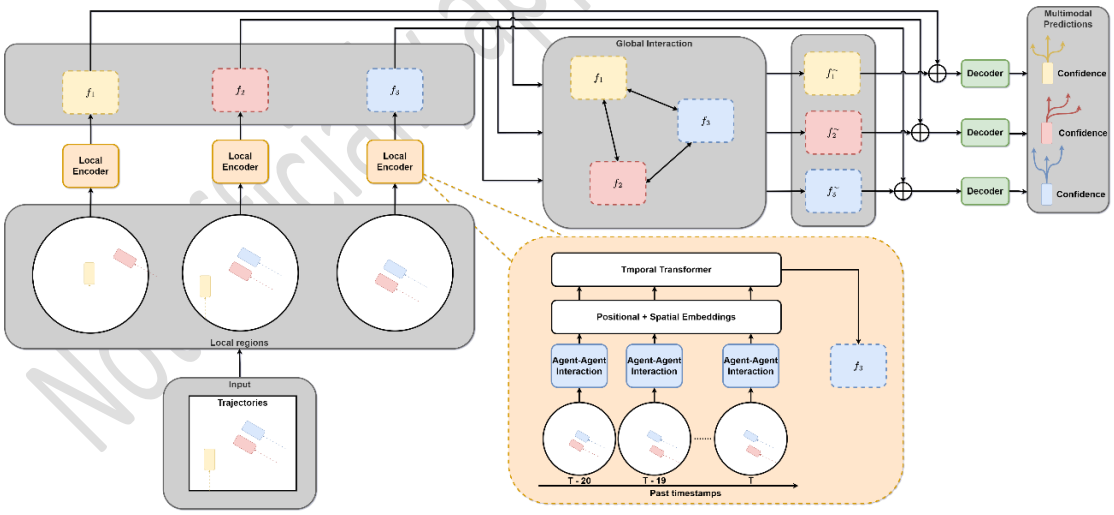


Figure 32: Framework proposal.

4.2.4 Metrics for evaluation

Most motion prediction datasets (either in the field of automated vehicles or others focused on pedestrian motion prediction) use the same metrics to evaluate the performance of the different proposed algorithms. We will use to evaluate our proposal the same metrics that Argoverse 1 [215] uses for its Leaderboard:

1. Minimum Average Displacement Error (MinADE): measures the minimum average Euclidean distance between the ground truth trajectory and the predicted trajectory over a specified prediction horizon.

$$\text{minADE} = \min_{i=1}^{K=N} \frac{1}{T} \sum_{t=1}^{T} \sqrt{\left(x_{i,t}^{pred} - x_{i,t}^{gt}\right)^2 + \left(y_{i,t}^{pred} - y_{i,t}^{gt}\right)^2}$$

where N is the total number of predictions or modes, T is the number of time steps, $(x_{i,t}^{pred}, y_{i,t}^{pred})$ are the predicted coordinates of vehicle i at time step t, and $(x_{i,t}^{gt}, y_{i,t}^{gt})$ are the ground-truth coordinates of vehicle i at time step t.

2. Minimum Final Displacement Error (MinFDE): calculates the minimum Euclidean distance between the final ground truth position and the final predicted position over the prediction horizon.

$$minFDE = min_{i=1}^{K=N} \sqrt{\left(x_{i,t}^{pred} - x_{i,t}^{gt}\right)^2 + \left(y_{i,t}^{pred} - y_{i,t}^{gt}\right)^2}$$

where N is the total number of predictions or modes, $(x_{i,t}^{pred}, y_{i,t}^{pred})$ are the predicted coordinates of vehicle i at time step t, and $(x_{i,t}^{gt}, y_{i,t}^{gt})$ are the ground-truth coordinates of vehicle i at time step t.

3. Miss Rate: This metric evaluates whether the predicted trajectory passes through a predefined area around the true trajectory. If the predicted trajectory does not enter this area, it is considered a miss.

4.2.5 Future work

This task has dependencies with other tasks that needs to be fulfilled to complete the task. The next list provides a brief description about the different works that are planned to be done:

1. For the implementation of the system in our framework we plan to use ROS2 Humble and CARLA 0.9.14 as simulation entity. Everything will be in its own containerized environment with all dependencies.

2. The complete proposal method will be deployed into the full perception framework in the EVENTS project, where inputs to the model are the detected and tracked agents.

3. The motion prediction will be enhanced and evaluated in a simulated V2X environment, where the ego-vehicle has an enhanced perception from different agents and infrastructures as well.

4. The output trajectories must be integrated into the path planning and decision-making tasks, synchronization with T4.1 and T4.2 is needed to integrate relevant information and validate the usefulness of the approach.

5. Study the integration of HD map information to improve predictions. Nevertheless, the current proposal, based on social interactions, achieves an excellent performance that allows it to be used in controlled environments. If the requirements of the project make it necessary to use HD-map, Lanelet2 will be used, which is a standard in autonomous research.

4.3 EXP3: Self-assessment and reliability of perception data with complementary V2X data in complex urban environments

EXP3 aims to showcase safe automated driving in complex urban environments with occlusion using onboard self-assessment methods and V2X data to integrate reliability assessment outputs into an onboard perception system. A more detailed description of EXP3 is proposed in Deliverable D2.1 “User and System Requirements for selected Use-cases” [35]. A system architecture for EXP3 is designed in Deliverable D2.2 “Full Stack Architecture & Interfaces” [36], which is a subset of the project’s master architecture. The architecture of EXP3, displayed in Figure 33, defines the internal data flow between the different modules besides the input and outputs. The architecture shows that for the self-assessment, a reliable base, including data pre-processing, object detection, and object tracking, is necessary. While the later work focuses on the self-assessment for object tracking, until now, some efforts have been made to create a reliable base, which includes the development of environment models to represent the vehicles surrounding. In the remainder of this chapter, first, an environment model for pedestrians is explained in detail. Then a flexible and adaptive grid map representation of the environment is presented as another approach for the environment state estimation. Third, an advanced Labeled Multi-Bernoulli Filter is described for a fast and robust implementation of the environment model using multi-sensor setups. Finally, the integration of self-assessment methods into the environment state estimation is outlined in an outlook section.

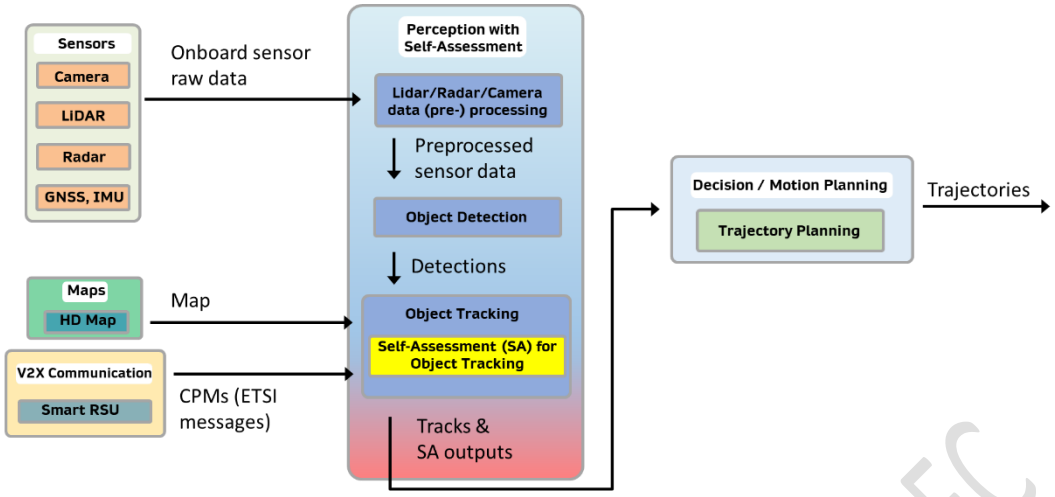


Figure 33: High-level Full Stack Architecture and Interfaces [36].

4.3.1 Pedestrian environment model

Automated driving technology has made significant progress in recent years thanks to advancements in perception and planning algorithms. A vehicle needs to fully recognize the surrounding environment to navigate safely. This can be achieved through different environment models such as grid maps and target lists. However, current environment models only display the location of other traffic participants and the drivable area, ignoring the unique characteristics of pedestrians. Unlike cars, pedestrians use gestures to communicate with other traffic participants, such as waving through a vehicle at a crosswalk or a police officer regulating traffic. Besides the pedestrians, also cyclists rely in their communication on hand signs, e.g., in case they want to turn. Unfortunately, current environment models lack important information regarding poses that can facilitate safe and adequate communication between automated vehicles and pedestrians. Therefore, an improved environment model is proposed that incorporates both the pedestrian's position and pose. The new pedestrian environment model enables gesture recognition, human behavior understanding, and body pose forecasting. This helps to provide a better understanding of pedestrians and other persons in urban scenarios. In the architecture in Figure 33, the pedestrian environment model can be used as a base for the self-assessment for object tracking when focusing on persons.

The proposed method only needs frames from a monocular camera sensor as well as data from a self-localization system to create the pedestrian environment model. A general overview of the approach is given in Figure 34. First, each camera frame is processed by a neural network for human pose estimation to get the 2D skeleton from each person displayed in the image. To extract the skeletons, the bottom-up human pose estimation approach CID [224] is used, which detects and extracts the keypoints of all persons on an image in one step.

In the next step, the extracted 2D skeletons are associated with tracks from earlier time steps to get a skeleton sequence for tasks following in the processing chain. Due to the ego-motion of the vehicle the detected skeletons and the tracked skeleton sequences might have no overlap in the image plane. Therefore, the ego rotation of the vehicle is compensated in the tracked skeleton sequences with data from the self-localization system and fits the location of the current detection. Following, the compensated skeleton sequences are associated with the newly detected skeletons. Here, the generalized intersection over union [225] and the Hungarian algorithm [226] are used for the association. This step is visualized in the lower left of Figure 34 in the *Person Tracking* part. One updated skeleton sequence per detected pedestrian is forwarded to the *World Position Estimation* step for further processing.

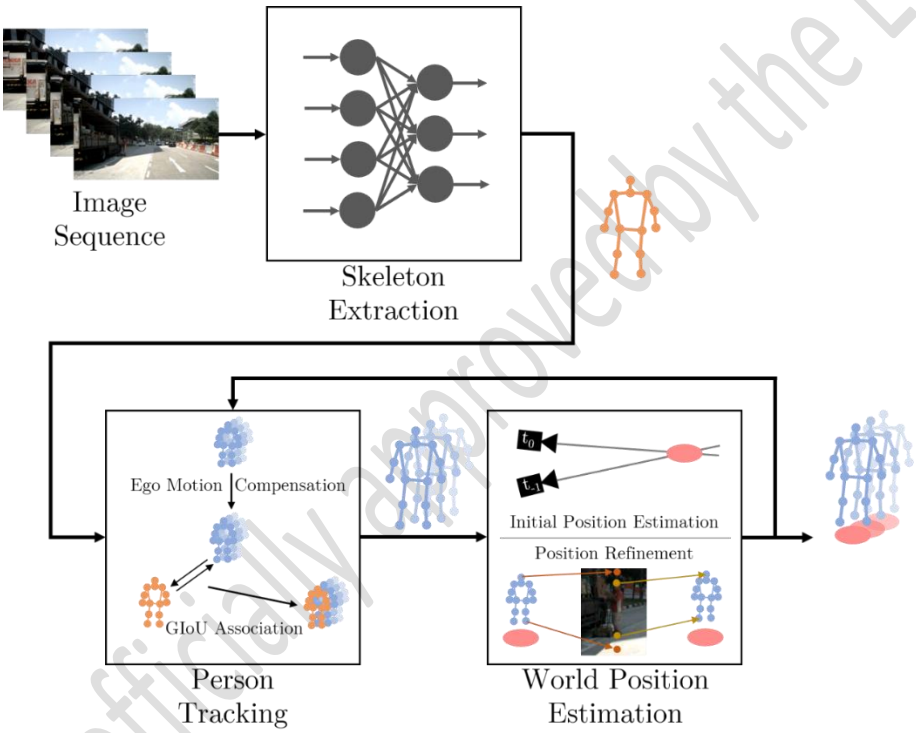


Figure 34: Overview of the pedestrian environment model [227].

The skeleton sequences contain only 2D coordinates in the image plane and no information about the pedestrians' location in the world coordinate system. The 3D position in the world coordinate system is determined in two steps: an initial position estimation and a refinement step.

1. In the initial position estimation step, geometric dependencies between two consecutive time steps are used for a first position estimate. Therefore, a 3D ray in the space is calculated with the camera parameters for each keypoint. Then, the closest point between the same keypoints of two consecutive frames is determined. The pedestrian's position is assumed as the mean point of all skeleton keypoints.

2. The refinement step uses the initial pedestrian's position and re-projects the pedestrian's height into the image plane. For the re-projection, a mean pedestrian's height of 1.7m is assumed. The re-projected pedestrian's height is compared with the original pedestrian's height. If both heights correspond, the distance between the pedestrian and the ego vehicle is assumed as correct; else, the distance is refined. Therefore, a refinement factor is calculated with the re-projected and the correct pedestrian's height, and the distance is improved with the factor. This is repeated until the re-projected height corresponds to the original height.

The method provides, after the three steps, for each pedestrian a position in the world coordinate system and a skeleton sequence. The skeleton sequence can used in the following tasks, e.g., action recognition or behavior understanding.

The proposed pedestrian environment model is evaluated on two datasets: the real-world dataset nuScenes mini-set-split [228] and a dataset simulated with the CARLA simulator [229]. On the nuScenes dataset, only the day sequences are used because the CID human pose estimation network is not trained on night-view images. For the simulated dataset, 2D skeletons are simulated with CARLA because, due to the domain gap, the human pose estimation neural network does not work on the simulated data. For the evaluation, on both datasets the absolute distance error in meters [m] between the estimated distance and the ground truth distance is calculated. Additionally, the distance error in relation to the ground truth distance is given as a relative error in percent [%].

On the simulated dataset, the absolute error is around 9.1m, with a mean distance between the vehicle and the pedestrian of 57m. This results in a relative error of around 17%. This result is comparable with the result on the nuScenes dataset, which has an absolute error of around 2.5m, corresponding to a relative error of 16% with a mean distance of 15m.

The presented pedestrian environment model can be used in EXP3 to model the pedestrians in the vehicles surrounding. In the next section, with the grid map, a more generalized environment representation method is presented, which can also display other traffic participants like vehicles.

4.3.2 Adaptive patched grid mapping

As another approach to model the environment for the perception of an autonomous vehicle to enable the state estimation in EXP3, we focus on a grid mapping representation of the environment. The main idea of grid mapping is to divide the environment, especially unstructured environments, into cells. These cells collect information about the state of the corresponding location. The most fundamental

type of cell information is the occupancy, which leads to occupancy grid maps, see, e.g., [230].

The field of grid mapping is an active research area, especially efficient grid mapping gets more and more interesting. There are already works about non-uniform grid and cell resolution [231] and a special patch structure to accelerate particularly the computation time [232]. One of the major challenges in this area so far is the flexibility of performing a grid map with online adaptable configurations. To this end, we develop and propose an Adaptive Patched Grid Mapping (APGM) approach [233], which enables a situational-aware grid-based perception. This perception approach allows not only a situational-dependent perception, but also a requirement-dependent perception. Moreover, it is a flexible representation of the surrounding unstructured environment and, most importantly, it allows to dynamically change external requirements. These can be the cell resolution, areas of interest, and horizon targets. All in all, the implemented APGM approach is also memory efficient.

4.3.3 Fast product multi-sensor labeled multi-Bernoulli filter

Another aspect of environmental state estimation is the development of fast and robust object tracking algorithms in order to integrate additional self-assessment methods into the tracking algorithms for EXP3. The computational complexity of a multi-sensor multi-object tracking algorithm can easily become quite high, especially in distributed sensor setups, as it is typically the case in infrastructure-based augmented perception. Therefore, implementing and running such algorithms for practical applications is a challenging task. Usually, there is a trade-off between tracking performance in terms of accuracy and computational complexity. For this purpose, we develop and propose the so-called Fast Product Multi-Sensor Labeled Multi-Bernoulli (FPM-LMB) filter [234]. This filter and tracking algorithm combines computational efficiency while obtaining robustness. This can be seen especially in challenging situations such as when unknown occlusions or sensor failures occur.

4.3.4 Outlook

In addition to building up the environment model and state estimation, this task is closely related to the self-assessment task in Task 3.5 of the project. Thus, we have already started working on the self-assessment of filtering and tracking algorithms, which is closely related to the environment state estimation in EXP3. Our approach is based on the work in [235],[236], which investigated the use of subjective logic for self-assessment in Kalman filtering. Building on this approach, in the scope of EXP3 we develop and propose a self-assessment framework for multi-sensor Kalman filters [237]. The developed self-assessment framework is capable of assessing nonlinear Kalman filters and additionally provides an overall assessment of the overall performance of the filtering algorithm. However, since the next deliverable D3.2 of

the EVENTS project focuses on the self-assessment of the perception system, we will not go into further details in this deliverable about the aforementioned self-assessment framework. We will provide more details of this self-assessment framework and the approach in D3.2 to achieve the goals of implementing a self-assessment module for EXP3.

4.4 **EXP4 & EXP5**: Roadworks, unmarked lanes, narrow roads and a jammed highway

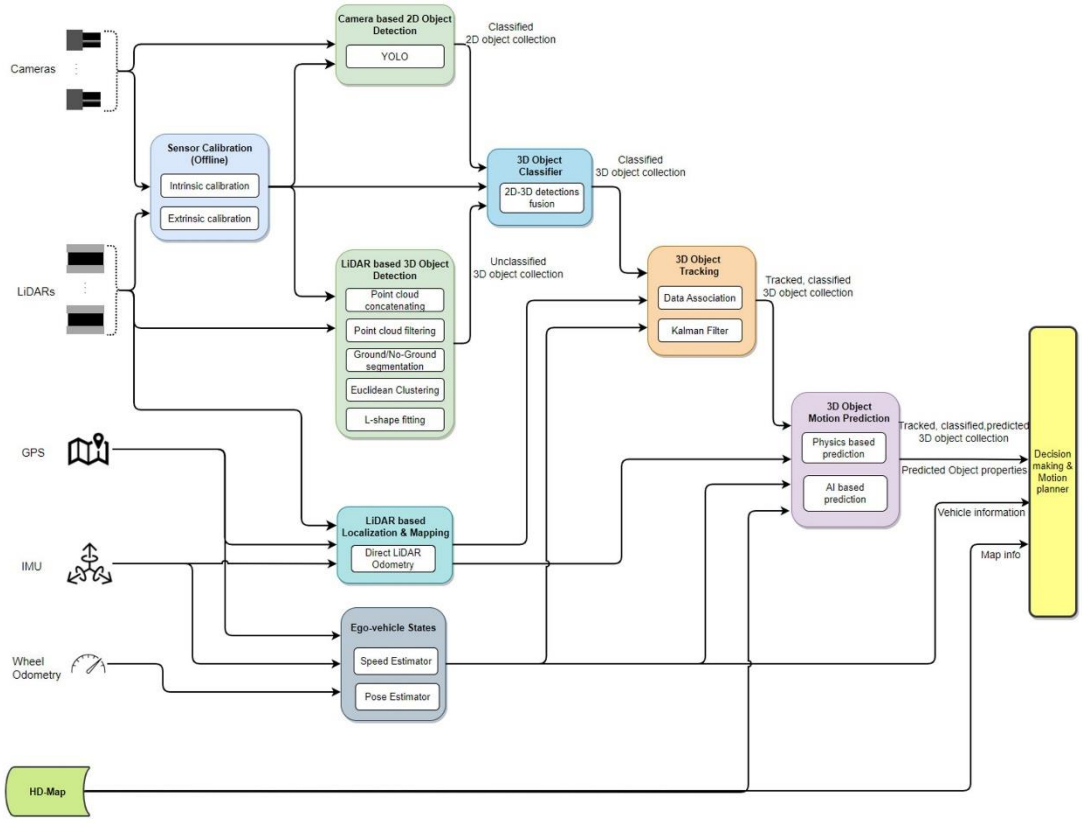


Figure 35 ***Planned perception architecture for EXP4 and EXP5.***

This section was contributed by project partner HIT.

4.4.1 State estimation

The environmental state estimation for both EXP4 and EXP5 is represented by the 3D object tracking module with Kalman filtering and data association algorithm as depicted in Figure 35. In this section, we present the 3D data association algorithm and Kalman filtering that we use to estimate the state of surrounding objects in the two experiments.

Since the objects are tracked from a moving vehicle it would be necessary to compensate for the movement of the ego-vehicle, based on its velocity/direction, prior to any data association and filtering activities. The proposed methodology bypasses this step by converting the detection to a map-centered coordinate system using the latest available pose of the ego-vehicle. To elaborate on this, by default the detections are vehicle-centered (D_i^v), meaning they are relative to the pose of the ego-vehicle, on the other hand the map-centered detections (D_i^m) are static with respect to the pose of the ego-vehicle.

3D Data Association: This step is responsible for matching old (D_m^m / T_m^m) detections/trackers with new (D_c^m, D_l^m) detections of the same object. This is achieved by computing the 3D Intersection-over-Union (3D-IoU) between the new detections and existing tracked objects (equivalent to old detections) in a full factorial manner. As illustrated in Figure 8 the 3D-IoU outputs a number between 0 and 1 depending on how close the 3D objects are placed e.g. 1 for fully matched, 0.5 for partially matched and 0 for not matching.

Having a 3D-IoU for each pair of new detections and existing tracked object, results in a combinatorial optimization problem which can be solved with the Hungarian method. To ensure the efficient solution of this optimization problem, a C++ program was developed using Google's OR-Tools. Practical real-world experiments have revealed latency levels of less than 20ms in congested urban environments proving the computational efficiency of the implementation for the purposes of the project.

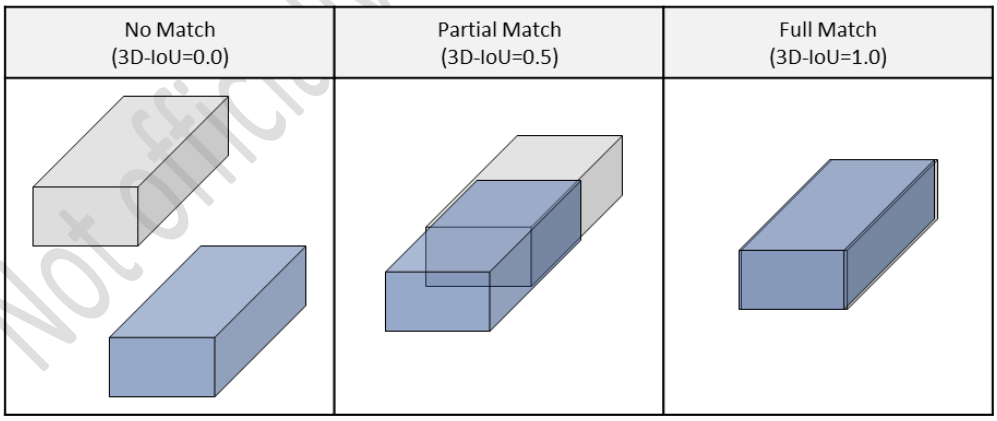


Figure 36: Illustrative example showing the different 3D-IoU results depending on the level of closeness between 3D objects

Birth/Death Memory: This sub-process is responsible to manage the actions taken either for new objects (D_u^m) entering the scene or for tracked objects (T_u^m) that might be leaving the scene.

In the case of tracked objects (T_u^m) which might be leaving the scene, instead of deleting them in the first instance that are not matched with any new detection, they are stored in memory for several iterations/frames. This is beneficial especially in cases where objects might be occluded thus not possible to be detected for few frames. However, if the previously tracked object remains unmatched for several consecutive frames, defined by the user, then it will be deleted from the memory permanently.

In the case of new detections (D_u^m) entering the scene, these are used to initiate new trackers. However, to avoid false positive tracked objects, the user can define a minimum number of matching events between new detections and newly created trackers. This is done to increase the tracking confidence before reporting them as tracked objects.

3D Kalman Filtering: Last step of the process is to filter out noise that might be originated from the object detection modules by using a constant velocity model. The model vector consists of 7 values:

- X coordinate of the object
- Y coordinate
- Heading (θ)
- Length
- Width
- X-velocity
- Y-velocity

The observation vector is made of 5 values: *x, y, θ, l, w.*

The model transition (*F*) is defined with the following matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & dt & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & dt \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The measurement matrix (*H*) has the following values:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Initial uncertainty P_0 and measurement uncertainty (*R*):

$$\begin{bmatrix} \sigma_{xyI}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{xyI}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\theta I}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{xyI}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{xyI}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{vI}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{vI}^2 \end{bmatrix} \quad \begin{bmatrix} \sigma_{xyM}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{xyM}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\theta M}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{xyM}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{xyM}^2 \end{bmatrix}$$

where $\sigma_{xyI} = 2$, $\sigma_{\theta I} = 0.88$, $\sigma_{vI} = 5$ for initial uncertainty, $\sigma_{xyM} = 0.5$, $\sigma_{\theta M} = 0.25$ for measurement uncertainty.

Process covariance matrix (Q) is defined as follows:

$$\begin{bmatrix} \sigma_{xyP}^2 & 0 & 0 & 0 & 0 & \sigma_{xyP}\,\sigma_{vP} & 0 \\ 0 & \sigma_{xyP}^2 & 0 & 0 & 0 & 0 & \sigma_{xyP}\,\sigma_{vP} \\ 0 & 0 & \sigma_{\theta P}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{lwP}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{lwP}^2 & 0 & 0 \\ \sigma_{xyP}\,\sigma_{vP} & 0 & 0 & 0 & 0 & \sigma_{vP}^2 & 0 \\ 0 & \sigma_{xyP}\,\sigma_{vP} & 0 & 0 & 0 & 0 & \sigma_{vP}^2 \end{bmatrix}$$

where

$$\sigma_{xyP} = a_{xy}\,\frac{dt^2}{2},\quad \sigma_{vP} = a_{xy}\,dt,\quad \sigma_{\theta P} = a_{\theta}\,\frac{dt^2}{2},$$

$$\sigma_{lwP} = 0.1\,m,\quad a_{xyP} = 4\ {}^{m}/_{s^2},\quad a_{\theta} = 5\ {}^{rad}/_{s^2}$$

4.4.2 Motion/Object Prediction

HIT plans to develop motion prediction model that can operate in the changing road configuration that can be found in EXP4 and EXP5. In particular, in EXP5, when road works are present, this can change the structure and behaviour of other road users. To this end, we seek to extend state of the art machine learning based approaches to handle the case when road works changes the configuration of the road. A summary of the planned work is as follows:

- Leverage state of the art open-source algorithms from big AD dataset challenges as a base.
- Extend approach to handle the scenario when road works are present, as such change in road structure will materially impact the predicted trajectories.

5. Augmented perception by V2X

5.1 Introduction

In this task, the general goal is to increase the field of view (FOV) of the connected and autonomous vehicle's (CAV's) onboard perception with external information by vehicle-to-everything (V2X) data. To this end, the accuracy and robustness of the CAV's onboard perception can be improved. In the scope of this task, the main connected experiments with the augmented perception by V2X are EXP2 and EXP3. In connection with EXP2 "Re-establish platoon formation after splitting due to roundabout" the goal is to develop a collective perception module, which is planned to be tested in a simulation environment. Moreover, in EXP3 "Self-assessment and reliability of perception data with complementary V2X data in complex urban environments", the goal is to implement the integration of the V2X data, in terms of cooperative perception messages (CPMs), into the onboard perception system of the ego vehicle in order to apply the self-assessment method in a next step. However, so far, the main work for this task has been done in the scope of EXP2, which will be explained in detail in the following.

5.2 EXP2: Re-establish platoon formation after splitting due to roundabout

Effective connectivity among agents is a critical factor in safely navigating a roundabout environment, especially when executing a platoon manoeuvre with multiple vehicles. CAVs and other vehicles equipped with communication capabilities will collaborate and share information to achieve this objective successfully.

Furthermore, in order to share the information, collective (or cooperative) perception (CP) is used. This is a multi-agent system [238] in which agents share perceptual information such as their state (e.g., vehicle position, pose, speed, acceleration), their tracked object list, or even their tracked objects' intentions which was first used in swarm robotics [239]. In the cooperative, connected, and automated driving field, CP enables CAVs to exchange driving environment perception data. In this case, V2X received information extends CAV's FOV beyond the on-board FOV, considerably improving CAV perception in Non-Line of Sight scenarios [240]. It also constitutes a redundant 'sensor' for the CAV within the on-board FOV. CP service is currently standardized by the European Telecommunications Standards Institute (ETSI) as a second-generation V2X communication service. In that context, CP has been studied more as an instantiation of a Vehicular Ad-hoc NETwork (VANET), via a combination of traffic/network simulation environments and targeting at answering questions such as up to how many connected agents per topology, communication protocols, latency and frequency of messages to be broadcasted, without studying the CP content generation (e.g. fusion techniques) and the assessment of the derived (collective)

object detection. In this work, the problem is cast as a multi-agent perception testing problem where the focus is on the shared information content fusion under the presence of occlusions and sensor measurement uncertainties (uncertainty propagation). CP network aspects are out of scope since we focus only on the perception layer.

5.2.1 Objectives and approach

Our strategy centers on utilizing V2X communication through WLAN technology to enable a synchronized manoeuvre through the exchange of data similar to the ETSI cooperative awareness message (CAM) and cooperative perception message (CPM) standards among different agents. This data exchange amplifies our awareness of the surrounding environment while ensuring a basic level of compatibility with ETSI standards. This approach enables us to validate coordination and algorithms within the use case without being constrained by the necessity for specialized V2X communication hardware.

To accomplish the platoon coordinated manoeuvre, and enhance the perception of the vehicles coordinated manoeuvre, a collective perception module will be deployed within the experiment. This module will elevate the confidence in our perception of the ego-vehicle by consolidating information from multiple vehicles.

The primary objective in this context is to support a de-centralized collective perception approach that can assist CAVs' urban decision making at roundabouts: the approach allows sparsely distributed agents to form a global view on a common spatially distributed problem without any direct access to global scene knowledge (i.e. where CAVs have no access to knowledge, for instance coming from a smart infrastructure node) and only based on a combination of locally perceived information [241]. The main question is how to share and combine the estimated information to achieve the most precise global estimate in the least possible time. From an algorithmic point of view, the method assumes a late fusion scheme (i.e. pre-processed object-level data like bounding boxes and confidence scores are shared) and employs a Bayesian logic which offers inherent metrics for assessing the quality of the method. CP module testing in simulation and in hybrid real-world & simulation environment is of interest.

Furthermore, the integration of the data coming from the CP module be closely aligned with T3.3, allowing us to seamlessly incorporate this expanded perception of the environment into our scene definition and motion prediction framework. This integration is crucial for achieving a comprehensive understanding of the driving scene and ensuring safe and efficient manoeuvres.

5.2.2 Simulation environment

Currently there is special focus towards getting the most accurate representations of the simulated environments used for EXP2. Hence, as the integration between CARLA Simulator platform [254] and ROS2 Humble framework [255] is crucial for this task, there is a requirement of developing further solutions on top of the actual implementations of the CARLA-ROS bridge. So far, TECN has achieved the following:

- Launch multiple vehicles with customized sensing capabilities within a CARLA-ROS2 bridge environment (LiDAR, GNSS, Camera). An example scene is shown in Figure 37.
- CARLA-ROS2 bridge: Integration of a library to enhance the current sensing given by CARLA. This library is called PCSim [253], which provides different realistic LiDAR models (Velodyne, Ouster, Robosense) for CARLA simulator. In both T3.3 and T3.4, it becomes necessary to implement more realistic LiDAR models, like the current implemented. This is caused since there are always concerns about extrapolating simulated data to real-world environments, and detection and motion prediction models trained in such environments may not perform as expected in the real world. Therefore, different pre-trained models in real-world datasets (KITTI, nuScenes) may be evaluated in this library, and vice versa.

Thanks to these developments, the developed perception stack can increase its scalability in V2X scenarios, either in detection, in obtaining perception from different external agents, or in motion prediction tasks.

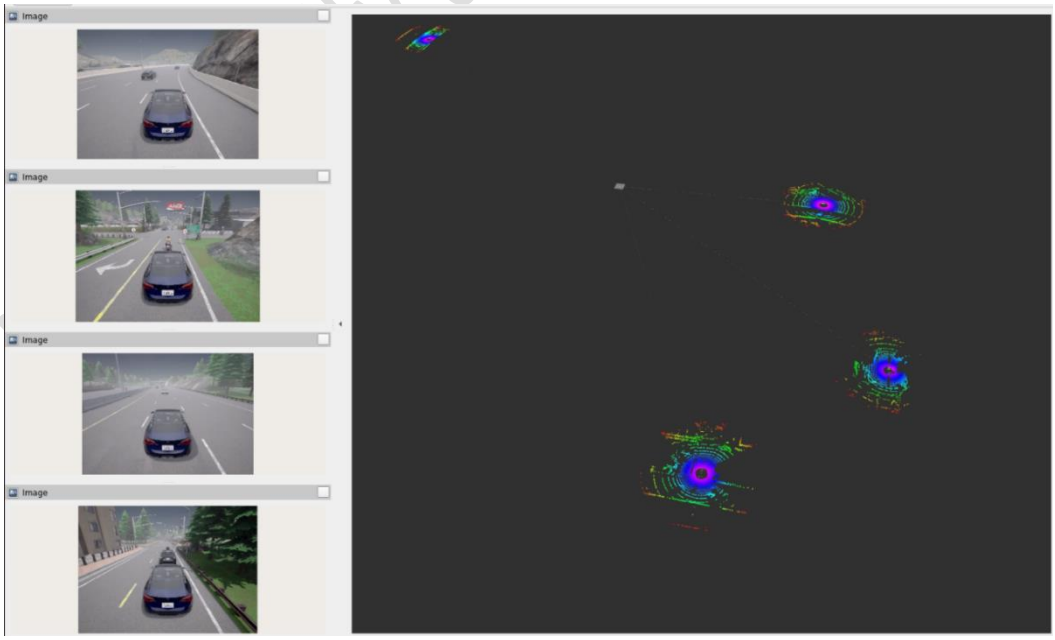


Figure 37: CARLA-ROS-Bridge with multiple vehicles in ROS2 Visualization Tool (RViz2).

5.2.3 Collective perception module

The CP module developed by ICCS is also part of EXP2 which was described in Deliverable D2.1 “User and System Requirements for selected Use-cases” [35]. EXP2 is categorized under the first Use Case (UC1) defined in the EVENTS project, which is concerned with safe and resilient automated driving in complex urban environments. The focus here is on urban roundabouts with a specific focus on vehicle-to-vehicle integration and advanced control based on CP. More specifically, in EXP2, “Re-establish platoon formation after splitting due to roundabout”, a coordinated platooning planning is investigated via V2X data integration. The logical scenario is as follows: A platoon ensemble composed of three CAVs (one CAV leader and two CAVs as followers) in an urban environment is assumed; the platoon is split because of traffic when approaching and crossing a roundabout (driving rules in the roundabout are assumed to prioritize the vehicles inside the roundabout). The followers should be able to reach the leading vehicle ensuring string stability also under curved trajectories. Planning of re-joining the platoon takes advantage of CPMs fused information (and confidence) that is made available to the follower when entering the roundabout (Figure 38).

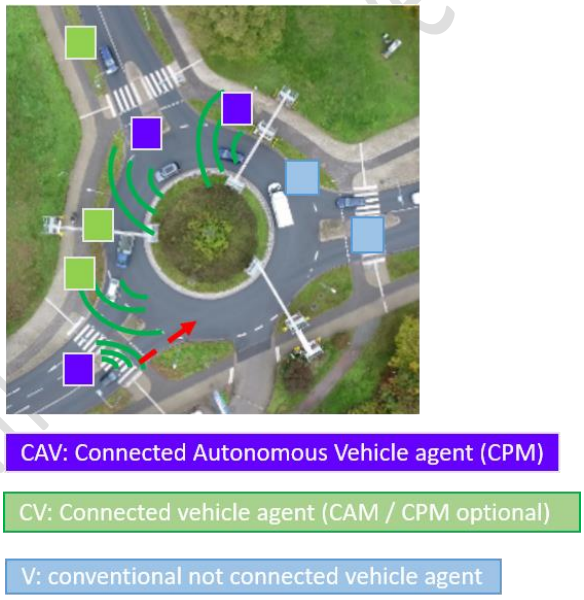


Figure 38: EXP 2 scene and type of agents.

5.2.3.1 Architecture

A system architecture for EXP2 is designed in Deliverable D2.2 “Full Stack Architecture & Interfaces” [36], which is a subset of the project’s master architecture. In Figure 39, a small variation of D2.2 EXP2 architecture is provided to better visualize the CP module role in the project’s master architecture.

In the following, a more detailed logical architecture of the CP module is derived. As shown in Figure 40, CP essentially replaces the subject automated vehicle perception by providing enhanced scene understanding. CPMs received from connected agents, sharing the same spatial area (in our case a roundabout area) with the subject CAV under test, are fused with CAV's local perception. The probabilistic certainty of collective object detection plus further consistency/plausibility checks between CP output and the (claimed) FOV/perceived object list of each connected (CCAV) agent will be utilized for assessing the reliability of the CP (this work will be reported as part of D3.2 as it is part of T3.5).

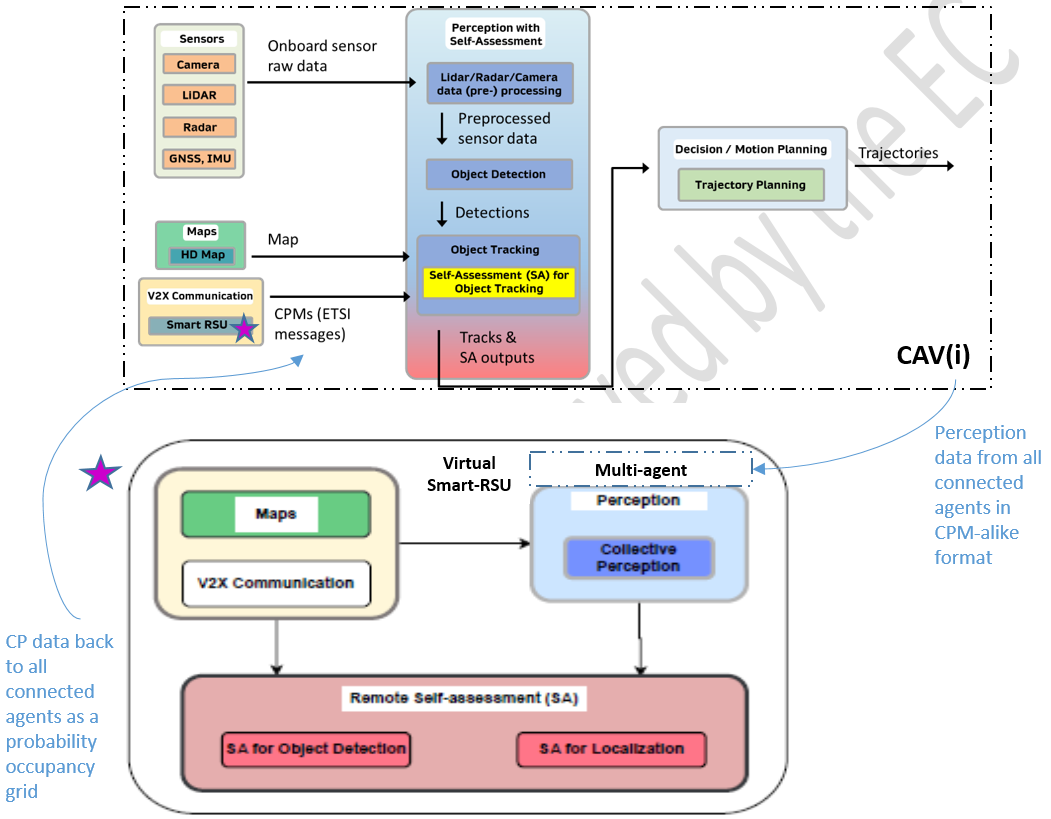


Figure 39: CP module in the project's reference architecture.

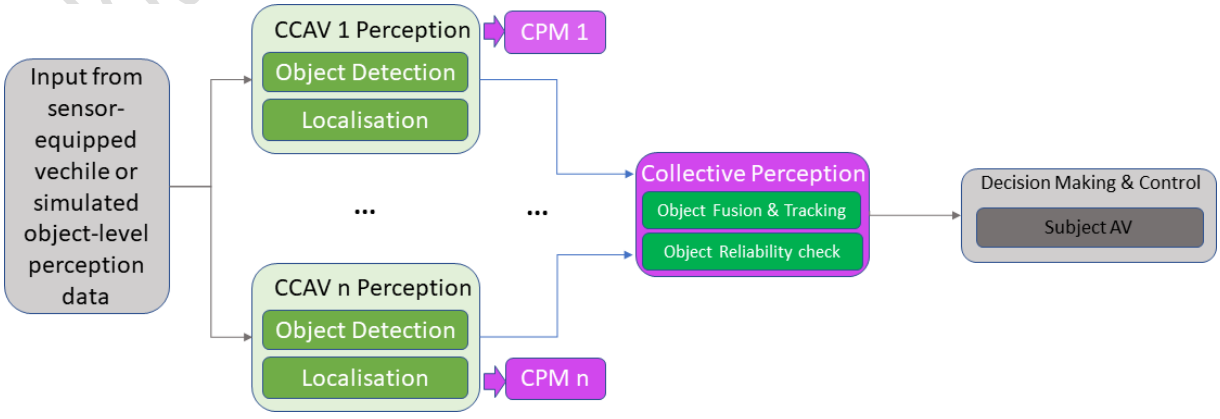


Figure 40: CP module high level components.

The CP module expects the following **input** from each CCAV present in the area of interest:

- **Ego FOV angle:** A single front camera implies a FOV angle equal to the FOV of the camera; more cameras and radar/LiDAR sensors may imply a larger FOV angle. In the figures below, a 360^o FOV angle is implied.
- **Ego state information:** This information consists of the
    1) position coordinates in x, *y* (valid also for CAM originators),
    2) speed vector v_x, v_y (valid also for CAM originators),
    3) acceleration vector a_x, a_y, and iv. Heading (yaw angle).
- **Observed object information:** This information concerns each one of the distinct objects observed by the CCAV and consists of the
    1) position coordinates in x, *y*,
    2) speed vector v_x, v_y,
    3) acceleration vector a_x, a_y, and
    4) heading (yaw angle).
- **Estimated free space:** This field is optional since it can be partially deduced from the observed objects' information.
- **Uncertainty of the measured values:** A quantification of the uncertainty of each measurement via, e.g., respective standard deviations.

Note: The inputs above are a subset of the ETSI CPM. However, soft ETSI compliance is considered achieved since the main information about objects and free area estimation have been considered in our CP implementation.

The **output** of the perception module of a mobile agent (in our case, a CCAV) has been commonly formalized in terms of a probabilistic occupancy grid or map [242][243][244][245][246] and this is the representation selected for CP module implementation. This is due to several inherent features of the particular approach that make it suitable for pertinent applications. These features include (i) the utilization of quite general and useful sensor models like the forward sensor model [242][245][247], (ii) the straightforward and intuitive Bayesian method for fusing observations derived from multiple such sensor models [242][247], plus the fact that its probabilistic nature lends itself for (iii) building a large variety of Bayesian filtering algorithms in top of it [244][246][248][249][250][251] and (iv) inherently deriving reliability metrics for the output, like entropy in [247].

Note: In our approach, the virtual road-side unit (RSU) node responsible for CP execution can deliver aggregated CPMs back to the connected vehicle agents. Additionally, a probabilistic occupancy grid of the observed scene is also returned to the connected agents.

5.2.3.2 Algorithmic approach

In this section, the formalization of the output of the CP module as a probabilistic occupancy grid is presented. The chosen representation exploits the straightforward and intuitive Bayesian method for fusing observations derived from multiple observers and has been very popular for solving multi-object tracking problems due to its inherent explainability properties leading to well-defined reliability metrics for the output [243][245].

5.2.3.2.1 Formalizations and problem statement

Consider a 2-dimensional bird's eye view of the area of interest; see e.g., the following figures. We discretize it using a rectangular grid of size NxN , whose cells are indexed by $i = 1, \dots, N^2$. To each cell i we associate the binary random variable $A_i \epsilon 0,1$ where $A_i = 1$ means "cell i is occupied" and $A_i = 0$ means "cell i is not occupied". A Probabilistic Occupancy Grid is essentially a collection of N^2 probabilities $P(A_i = 1)$, $i = 1, \dots, N^2$, each one indicating the probability with which the corresponding cell i is occupied.

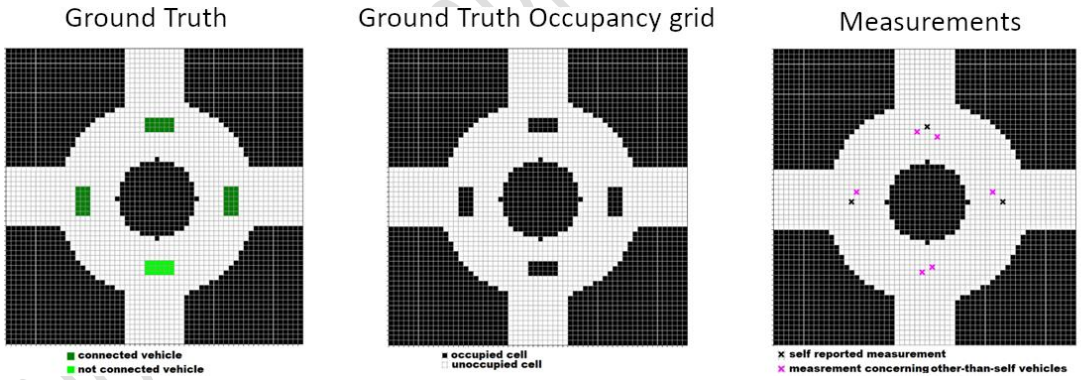


Figure 41: The ground truth with and without an occupancy grid and the measurements

Let the area of interest be a roundabout like the one depicted Figure 41. We assume the presence of both connected and not connected vehicles. Connected vehicles have the capability to share information concerning (i) their current individual heading, position, speed and acceleration, (ii) the presence or absence of objects within their respective individual FOVs, and (iii) heading, position, speed and acceleration for each of the perceived objects, in line with the description of inputs in Section 5.2.3.1. The ground truth of the scene is a non-random occupancy grid, where we know exactly which cells are occupied or not (middle illustration Figure 41). We are now ready to properly state the problem of interest:

Problem statement: Provided the observations and measurements of each individual CCAV and their statistical uncertainties/errors, how can we estimate the ground truth in terms of a probabilistic occupancy grid?

5.2.3.2.2 Algorithmic overview

The proposed algorithm consists of the following three equally important steps that are schematized together in the SW architecture diagram of Figure 42.

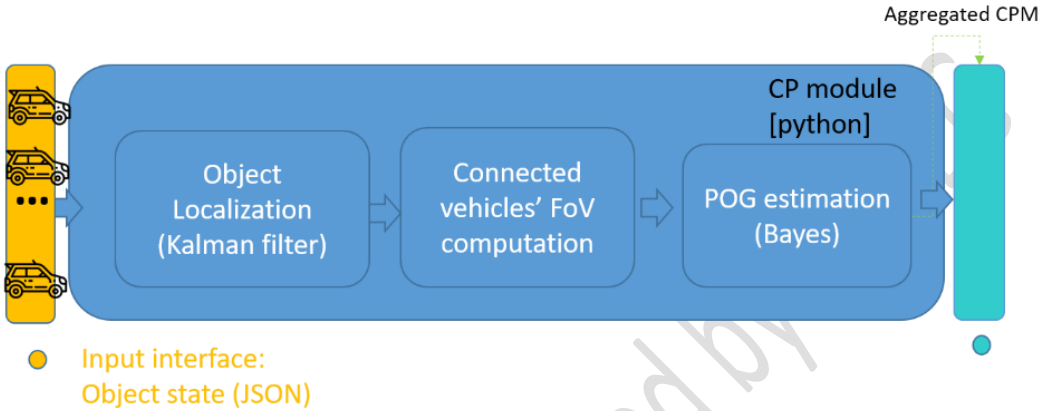


Figure 42: CP module's algorithmic sub-modules.

Step 1: The first step aims at the localization of reporting agents (in our case, CCAVs). For each distinct CCAV, we apply a Kalman filter considering only its self-reporting measurements. The rationale for excluding other measurements is that self-reporting measurements are based on differential GPS for location and on-board sensors for velocity, acceleration, and heading, which can be considered more trustworthy. Apart from that, initial experiments show that clustering measurements around centroids corresponding to unknown guessed vehicle positions can be extremely error-prone due to the unknown number of present vehicles and noisy measurements, leading to unstable estimations.

Step 2: The second step aims at estimating the FOV of each CCAV. Each CCAV is placed in its estimated (by step 1) grid position. Each other-than-self measurement (other objects detected around) obtained by CCAV is translated according to the CCAV estimated position and placed in the grid. Subsequently, a custom, GPU-implemented algorithm calculates the FOV of the CCAV, i.e., the grid cells for which the CCAV can provide information on their occupancy status (Figure 43 & Figure 44). Estimated FOVs may be eroded to account for measurement uncertainties.

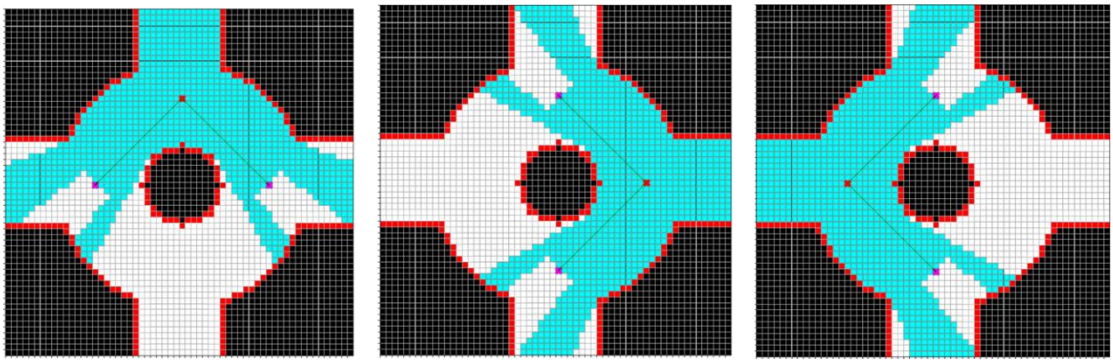


Figure 43: The CCAV's FOV.

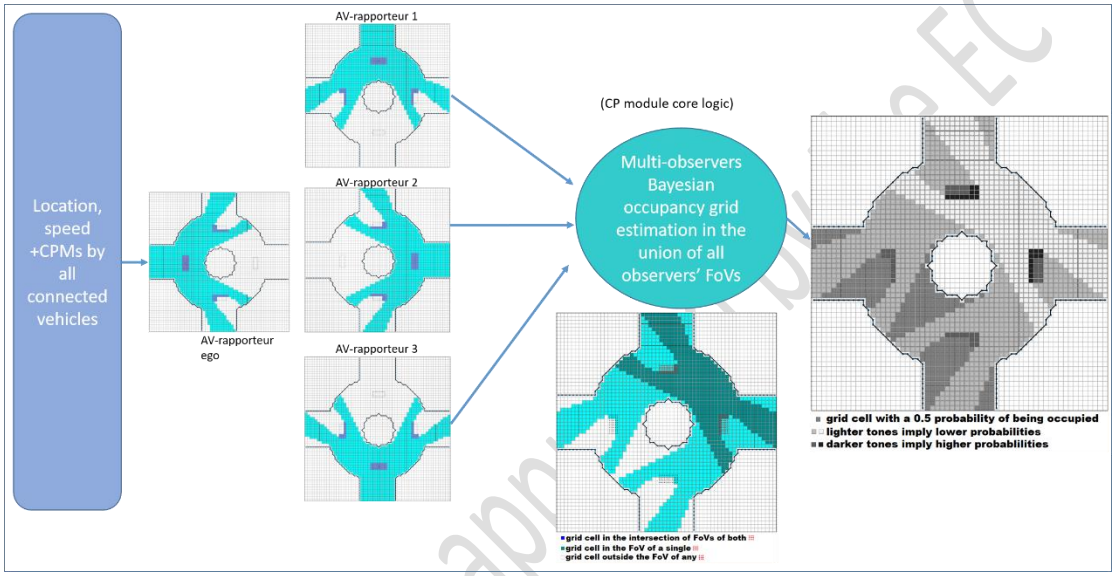


Figure 44: CP module logic and data flow focusing on the outputs from step 2 (FOV of each rapporteur AV and fused FOV) & and step 3 (POG: probabilistic occupancy grid).

Step 3: The third step aims at fusing the observations of each CCAV to form a probabilistic estimation for the occupancy grid of the entire area of interest. We assume a known individual perception model for each CCAV, provided in terms of the 4 probabilities $P(M_i = 0|A_i = 0)$, $P(M_i = 1|A_i = 0)$, $P(M_i = 0|A_i = 1)$, $P(M_i = 1|A_i = 1)$ of the standard forward sensor model [238][241], where $A_i \epsilon \{0,1\}$ is the random variable "cell i is truly occupied $(A_i = 1)$ or not $(A_i = 0)$" $M_i \epsilon \{0,1\}$ is the random variable "cell i is perceived as occupied $(M_i = 1)$ or not $(M_i = 0)$"

With the above probabilities and the observations of each particular CCAV regarding the occupancy state of each cell within the CCAV's FOV, we can take into account all observations regarding the occupancy state of any particular cell by applying Bayes rule recursively. Specifically, let M_i^k be the occupancy state of grid cell i as perceived by the agent k (CCAV). For $k = 1$, we have the standard Bayesian update

$$P(A_i = 1|M_i^1) = \frac{P(M_i^1|A_i = 1)P(A_i = 1)}{P(M_i^1|A_i = 1)P(A_i = 1) + P(M_i^1|A_i = 0)P(A_i = 0)}$$

$$P(A_i = 0|M_i^1) = \frac{P(M_i^1|A_i = 0)P(A_i = 0)}{P(M_i^1|A_i = 1)P(A_i = 1) + P(M_i^1|A_i = 0)P(A_i = 0)}$$

where $P(A_i = 1|M_i^1), P(A_i = 0|M_i^1)$ are posterior occupancy probabilities of grid cell i conditioned on M_i^1. The initial prior occupancy probabilities $P(A_i = 1),\ P(A_i = 0)$ are assumed to be equal to 0.5 to reflect the unknown occupancy state of cell i. Provided with k independent observations $M_i^1,\ M_i^2,\dots, M_i^k$ by k different agents (CCAVs) on the occupancy state of grid cell i, one can show the recursion

$$P(A_i = 1|M_i^1, \dots, M_i^k) = \frac{P(M_i^k|A_i = 1)P(A_i = 1|M_i^1, \dots, M_i^{k-1})}{P(M_i^k|A_i = 1)P(A_i = 1|M_i^1, \dots, M_i^{k-1}) + P(M_i^k|A_i = 0)P(A_i = 0|M_i^1, \dots, M_i^{k-1})}$$

$$P(A_i = 0|M_i^1, \dots, M_i^k) = \frac{P(M_i^k|A_i = 0)P(A_i = 0|M_i^1, \dots, M_i^{k-1})}{P(M_i^k|A_i = 1)P(A_i = 1|M_i^1, \dots, M_i^{k-1}) + P(M_i^k|A_i = 0)P(A_i = 0|M_i^1, \dots, M_i^{k-1})}$$

where $P(A_i = 1|M_i^1, \dots, M_i^{k-1}),\ P(A_i = 0|M_i^1, \dots, M_i^{k-1})$ are the "prior" occupancy probabilities of probabilities of grid cell i conditioned on $M_i^1, \dots, M_i^{k-1}$ and $P(A_i = 1|M_i^1, \dots, M_i^k),\ P(A_i = 0|M_i^1, \dots, M_i^k)$ are the posterior occupancy probabilities of grid cell i conditioned on $M_i^1, \dots, M_i^k$.

Hence, when a cell belongs in the intersection of several CCAV FOVs, the above calculations enable the joint consideration of all respective individual observations regarding its occupancy state, ultimately providing a probabilistic estimate constituting the *collective* perception of the CCAVs.

A fourth step can be added in order to provide a short-range prediction of the occupancy grid within the next timesteps of the experiment.

Step 4 (optional): Step 4 aims at tracking the timely evolution of the occupancy grid. There is a variety of proposed methods to approach this, each one with its merits and disadvantages [35][240][243][244]. However, for all of these methods, the basic heuristic idea is common. In each time step, instead of simply setting the initial prior probabilities $P(A_i = 1),\ P(A_i = 0)$ equal to 0.5 to reflect the unknown occupancy state of each cell, knowledge regarding the previous time step is considered. Specifically, the cells of the previous step occupied with high probability are moved according to corresponding velocity measurements to new cells, and the resulting predicted occupancy grid is used to derive the priors for the current time step [248].

5.2.4 Outlook and future work

In addition to building up the CP virtual module, this task is closely related to the self-assessment task in Task 3.5 of the project. Thus, we have already started working on the self-assessment of localization and object detection CP outputs, which depend on

the environment state estimation that the CP module receives as input. In our approach, we will focus on the online self-assessment of the module output reliability based on the probabilistic occupancy grid properties (this renders the self-assessment work of T3.5 closely linked with the implementation work of T3.4). Since this work is still in the exploration phase (to be reported in D3.2), in this section, we briefly mention some examples of such indicators. A straightforward and intuitive indicator of the reliability of the output is the set of covariance matrices of the current Kalman filter recursions. For example, differential GPS systems provide measurements with errors in the order of 20-30 cm. Thus, position variance estimations above the order of 20-30 cm can be considered problematic. Both the percentage and the contiguity of grid cell regions with high confidence occupancy values (i.e., very high or very low occupancy probabilities) can also provide a reliability indicator for the output. Such indicators may also be applied locally, i.e., to quantify uncertainties in current critical regions of interest, like regions to be occupied in the immediate future. Other indicators concerning the consistency of claimed observations are under investigation based on the reliability and uncertainty propagation literature.

The module will be implemented with real-time operation capabilities by applying dedicated GPU code execution in Python environment. A hybrid validation framework will be used to understand limits and benefits (on the perception level) of an ETSI-alike CPM-enabled CP, under assumptions of different object detection uncertainties generated in simulation and obtained from real world, under different operational domain conditions. First results will be obtained in a first Python-based proof-of-concept. Then, the module will be dockerized and integrated with CARLA simulation environment via ROS-bridge. Finally, CARLA setup and scenario will be adapted in order to receive and visualize real CPM-alike data from an EVENTS prototype vehicle deployed in TECN test track. Evaluation will follow the methodology produced in T6.1 of the project.

When looking at the future work in connection with the simulation environment and the overall framework, the work is intended to be further expanded in the next iteration of this deliverable, task T3.4 and T3.3 are closely related and require integration with modules from the EVENTS ecosystem which will begin during the integration phase in simulation and vehicle platforms in WP5. A list of future works in regard to T3.4 is mentioned to be specified in the next iteration.

1) A message structure for CPM and CAM similar messaging using WLAN, by leveraging ROS2 and exchanging ego-vehicle and obstacle information.
2) Specialization considering the specific environment definition for each experiment, roundabouts (EXP2).
3) Integration of free space information provided by collective perception and self-assessment to improve the overall perception provisioning from external sources.

4) Leverage both detection and motion prediction within V2X scenarios using the current effort with CARLA-ROS bridge.

6. Conclusions

This Deliverable 3.1 reported on progress within tasks T3.1 – T3.4 of Work Package WP3 of the EVENTS project at month 16. The aim of this work package is to provide the machine perception system components needed to facilitate the various experiments (EXP1-EXP8) specified by the EVENTS project partners.

Task T3.1 involved the acquisition and adaptation of training data needed for the before-mentioned machine learning-enabled perception systems. A variety of existing public datasets for vehicle-based environment perception were explored, specifically, for road user object detection (TU Delft), collective perception (ICCS), traffic sign detection (HIT), and vehicle motion prediction (TECN). Some datasets were selected for WP3 perception tasks by the project partners (nuScenes and Waymo Open Dataset by TU Delft, Argoverse by TECN, Mapillary by HIT). Work was done on harmonization of annotations across datasets (ICCS), allowing unified access and ease-of-use. For those tasks, where no suitable datasets existed, work on sensor placement and calibration was reported (HIT). Finally, the road debris problem was introduced, and the collection of a new road debris dataset was described (APTIV).

Task T3.1 also involved the use of data-efficient techniques. The Deliverable focused on data augmentation based on existing real-world data (traffic signs, by HIT), and on data generation, either using machine learning models (GANs for bad weather, by ICCS) or from simulation (CARLA simulator, by ICCS). Apart from data augmentation, this Deliverable covered unsupervised learning, where 3D objects are learned automatically, without need for manual annotations from monocular camera and LiDAR data (TU Delft).

For task T3.2 on semantic scene analysis and precise localisation project partners are leveraging both novel sensors and algorithms to overcome the challenges as presented by the use cases. The novel sensors include 4D radars to semantically perceive the vehicles surroundings as well as to localise in poor weather. Furthermore, there is also a focus on developing methods that can overcome challenges that arise under degraded GNSS conditions using LIDAR based SLAM approaches.

Task 3.3. involves work on the integration of past and current measurements from on-board sensors to obtain the current environment state (incl. that of all relevant road users). Furthermore, it involves a prediction of how the environment state will evolve over time. Related to EXP2, this Deliverable reported on the prediction of vehicle movements and the behavior of dynamic obstacles at a roundabout. A Hierarchical Vector Transformer was adapted to a map-free model that employs social interaction to

compute multimodal predictions. Related to EXP3, this Deliverable covered an environment model for pedestrians, a flexible and adaptive grid map representation of the environment, and an advanced Labeled Multi-Bernoulli Filter for a fast and robust implementation of the environment model using multi-sensor setups (UULM). Related to EXP4 and EXP5 (Roadworks, unmarked lanes, narrow roads and a jammed highway) work on this task covered Kalman Filter-based state estimation.

Task 3.4 of augmented perception by V2X is mainly addressed by EXP2 and EXP3. The goal of this task is to extend the on-board perception of the ego CAV with information coming from other CCAVs or infrastructure sensors. Therefore, the information exchange between the vehicle and the external sensors is based on the ETSI standard. A scenario in this context is a coordinated platooning maneuver at a roundabout, where the focus is on a late fusion scheme in the presence of occlusions and sensor measurement uncertainties. First steps are done in simulation, using the CARLA simulator and ROS2 Humble.

This D3.1 described work in progress on T3.1-3.4. The final status will be reported in D3.2 submission among the topics within T3.1-3.4 that will be addressed till then:

- Validated camera data augmentation techniques and improved scenario data generation from simulations
- Improved 3D road user detection using the proposed unsupervised learning techniques, compared to a method that has only a limited amount of manual annotations available. Development of VRU motion prediction techniques in complex urban environments. Quantitative performance analysis, possibly on further datasets, such as EuroCity Persons 2.0 and View-of-Delft (TUD, EXP1).
- Prediction of vehicle movement at roundabout will be enhanced and evaluated in a ROS2 Humble and CARLA simulation environment. The case will be considered where the ego-vehicle has an enhanced perception based on fusing information from different agents and infrastructure on a probabilistic occupancy grid. Metrics for collective perception self-assessment will be also proposed and validated. Finally, the integration of HD map information to improve the predictions will be considered (TECN, EXP2).
- The on-board perception of the CAV will be augmented by V2X data in the form of CPMs from the UULM infrastructure pilot site to improve the overall perception towards safety and increased reliability (UULM, EXP3).
- Improved traffic sign detection, in particular the rare traffic sign classes using data augmentation. Vehicle motion prediction when the road layout has changed due to road work (HIT, EXP4 and EXP5)

- The road debris dataset will be extended to adverse weather and the debris height estimation module will be finetuned and quantitatively evaluated (APTIV, EXP6)

- A radar network will identify “ghost” reflections and will distinguish dynamic vs. static ones. This is preprocessing to a subsequent, purely radar based ego-motion estimation. Furthermore, the radar network will also assign a class probability for each radar target (EXP8, PERCIV)

References

[1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354–3361.

[2] 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 Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11621–11631.

[3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine et al., “Scalability in perception for autonomous driving: Waymo open dataset,” in Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2446–2454.

[4] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3d tracking and forecasting with rich maps,” in Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8748–8757.

[5] M. Braun, S. Krebs, F. B. Flohr, and D. M. Gavrila, “EuroCity Persons: A novel benchmark for person detection in traffic scenes,” IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 41, no. 8, pp. 1844–1861, 2019.

[6] T. Yau, S. Malekmohammadi, A. Rasouli, P. Lakner, M. Rohani, and J. Luo, “Graph-sim: A graph-based spatiotemporal interaction modelling for pedestrian action prediction,” in IEEE Int. Conf. on Robotics and Automation (ICRA), 2021, pp. 8580–8586.

[7] A. Palffy, E. Pool, S. Baratam, J. F. P. Kooij, and D. M. Gavrila, “Multiclass road user detection with 3+1d radar in the view-of-delft dataset,” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4961–4968, 2022.

[8] A. Milan, L. Leal-Taix´e, I. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” arXiv:1603.00831 [cs], 2016, arXiv: 1603.00831. [Online]. Available: http://arxiv.org/abs/1603.00831.

[9] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, “Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction,” in Proceedings of the IEEE/CVF Int. Conf. on Computer Vision, 2019, pp. 6262–6271.

[10] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, “Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior,” in Proceedings of the IEEE Int. Conf. on Computer Vision Workshops, 2017, pp. 206–213.

[11] G. Singh, S. Akrigg, M. Maio, V. Fontana, R. Alitappeh, S. Khan, S. Saha, K. Jeddisaravi, F. Yousefi, J. Culley, T. Nicholson, J. Omokeowa, S. Grazioso, A. Bradley, G. Gironimo, and F. Cuzzolin, “ROAD:The Road Event Awareness Dataset for Autonomous Driving,” IEEE

Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 45, no. 01, pp. 1036–1054, 2023.

[12] V. S. Saravanarajan, R.-C. Chen, and L.-S. Chen, "Lidar point cloud data processing in autonomous vehicles," in Int. Conf. on Electrical, Computer and Communication Technologies (ICECCT), 2021, pp. 1–5.

[13] S. Malla, B. Dariush, and C. Choi, "Titan: Future forecast using action priors," Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 11 183–11 193, 2020.

[14] B. Liu, E. Adeli, Z. Cao, K. H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal Relationship Reasoning for Pedestrian Intent Prediction," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3485–3492, 2020.

[15] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2636–2645.

[16] M. Braun, S. Krebs, F. Flohr and D. M. Gavrila. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 41, nr. 8, pp. 1844-1861, 2019.

[17] P. Cong, X. Zhu, F. Qiao, Y. Ren, X. Peng, Y. Hou, L. Xu, R. Yang, D. Manocha, and Y. Ma, "STCrowd: A multimodal dataset for pedestrian perception in crowded scenes," in Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 19608-19617

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

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

[20] B. Shuai, A. Bergamo, U. Buechler, A. Berneshawi, A. Boden, and J. Tighe, "Large scale real-world multi-person tracking," in European Conf. on Computer Vision (ECCV), 2022, pp. 504–521.

[21] O. Styles, V. Sanchez, and T. Guha, "Multiple object forecasting: Predicting future object locations in diverse environments," in Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 690–699.

[22] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, Felix Heide, Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. arXiv:1902.08913v3

[23] https://github.com/marcelsheeny/radiate_sdk

[24] http://cadcd.uwaterloo.ca/

[25] https://gazebosim.org

[26] http://wiki.ros.org/rviz

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros, Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004v3

[28] Ming-Yu Liu, Thomas Breuel, Jan Kautz, Unsupervised Image-to-Image Translation Networks. arXiv:1703.00848v6

[29] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, Eli Shechtman, Toward Multimodal Image-to-Image Translation. arXiv:1711.11586v4

[30] Xun Huang, Ming-Yu Liu, Serge Belongie, Jan Kautz, Multimodal Unsupervised Image-to-Image Translation. arXiv:1804.04732v2

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

[32] https://github.com/NVlabs/imaginaire/tree/master/projects/munit#hardware-requirement

[33] Wang, Ting-Chun et al. “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017): 8798-8807.
[34] 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.
[35] EVENTS Deliverable D2.1: User and system requirements for selected use cases (2023).
[36] EVENTS Deliverable D2.2: Full Stack Architecture & Interfaces (2023)
[37] “NCSA Tools, Publications, and Data,” NHTSA, [Online]. Available: https://cdan.dot.gov/. [Accessed October 2023].
[38] “Traffic Safety Facts Annual Report Tables,” NHTSA, [Online]. Available: https://cdan.dot.gov/tsftables/tsfar.htm. [Accessed 11 October 2023].
[39] “Mobility & Transport - Latest key figures,” European Commision, [Online]. Available: https://road-safety.transport.ec.europa.eu/european-road-safety-observatory/data-and-analysis/latest-key-figures_en#collision-matrix. [Accessed 11 October 2023].
[40] “ROAD TRAFFIC FATALITIES,” European commision, [Online]. Available: https://road-safety.transport.ec.europa.eu/system/files/2022-08/road_traffic_fatalities_in_the_eu_in_2020_total.pdf. [Accessed 11 October 2023].
[41] “Lawsuits for Accidents Caused by Road Debris,” Shouse California Law Group, [Online]. Available: https://www.shouselaw.com/ca/personal-injury/car-accident/road-debris-accident-lawsuit/. [Accessed 11 October 2023].
[42] “Look Out! 10 Most Common Types of Road Debris,” Geroge Sink, P.A. Injury Lawyers, March 2013. [Online]. Available: https://www.sinklaw.com/blog/look-out-10-most-common-types-of-road-debris/. [Accessed 11 October 2023].
[43] “About AAA Foundation for Traffic Safety,” AAA Foundation for Traffic Safety, [Online]. Available: https://aaafoundation.org/about/. [Accessed 11 October 2023].
[44] 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.
[45] “How Common Is Car Damage from Road Debris?,” Injury Law PLLC Todd W.Burris Attorney, 19 November 2021. [Online]. Available: https://www.toddwburrislaw.com/how-common-is-car-damage-from-road-debris/. [Accessed 11 October 2023].
[46] “Crash Investigation Sampling System,” NHTSA, [Online]. Available: https://www.nhtsa.gov/crash-data-systems/crash-investigation-sampling-system. [Accessed 11 October 2023].
[47] B. Zhang, G. Li, Q. Zheng, X. Bai, Y. Ding and A. Khan, “Path planning for wheeled mobile robot in partially known uneven terrain,” Sensors, vol. 22, no. 14, p. 5217, 2022.
[48] T. L. M. N. D. H. V. K. B. Gabriel Günter Waibel, “How Rough Is the Path? Terrain Traversability Estimation for Local and Global Path Planning,” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2022.
[49] A. C. a. H. Hirschm¨uller, “Stereo Camera Based Navigation of Mobile Robots on Rough Terrain,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.
[50] Y. Zhou, Y. Huang and Z. Xiong, “3D Traversability Map Generation for Mobile Robots Based on Point Cloud,” in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2021.

[51] T. Guan, Z. He, R. Song, D. Manocha and L. Zhang, “TNS: Terrain Traversability Mapping and Navigation System for Autonomous Excavators,” 2022.
[52] K. Zhang, Y. Yang, M. Fu and M. Wang, “Traversability Assessment and Trajectory Planning of Unmanned Ground Vehicles with Suspension Systems on Rough Terrain,” Sensors, 2019.
[53] M. K. C. ALEXANDER AXELSSON, “A Study on the Impact of Traversability Measures on Rough Terrain Path Planning,” Master Thesis KTH, 2020.
[54] R. Counts, “Off-Road Buying Guide,” edumuns, 26 March 2020. [Online]. Available: https://www.edmunds.com/vehicles/off-road-buying-guide/#:~:text=Generally%2C%20all%2D%20or%20four%2D,more%20should%20be%20pretty%20good.. [Accessed 12 October 2023].
[55] “What is the Difference Between Off-Road and On-Road?,” off-road handbook, 27 January 2022. [Online]. Available: https://offroadhandbook.com/what-is-the-difference-between-off-road-and-on-road/?utm_content=cmp-true. [Accessed 13 October 2023].
[56] “Speed Humps vs. Speed Bumps,” MaineDOT, [Online]. Available: https://web.archive.org/web/20130627171125/http://www.maine.gov/mdot/csd/mlrc/technical/shsb.htm. [Accessed 13 October 2023]. J. J. Fazzalaro, “SPEED BUMPS AND SPEED HUMPS,” Office of Legislative Research, 22 September 2006. [Online]. Available: https://www.cga.ct.gov/2006/rpt/2006-r-0567.htm. [Accessed 13 October 2023].
[57] “What’s the Deal With Speed Bumps?,” Sun Devil Auto, [Online]. Available: https://www.sundevilauto.com/whats-the-deal-with-speed-bumps/. [Accessed 13 October 2023].
[58] G. Watts, “Road humps for the control of vehicle speeds,” Transport and Road Research Laboratory, Crowthorne, Berkshire, 1973.
[59] B. Zheng, Z. Hong, J. Tang, M. Han, J. Chen and X. Huang, “A Comprehensive Method to Evaluate Ride Comfort of Autonomous Vehicles under Typical Braking Scenarios: Testing, Simulation and Analysis.,” Mathematics, vol. 11, no. 2, 2023.
[60] E. Lima, M. Silveira, J. A. Júnior, C. A. S. Carvalho, M. F. L. Antunes and D. M. Junqueira, “Evaluation of Vehicular Discomfort Measures Produced by Speed Bumps Using Numerical Simulation,” Journal of Mechanical Engineering and Automation, vol. 5, no. 4, pp. 113-123, 2015.
[61] “Advanced Radars. Aptiv's most advanced sensors push the envelope,” Aptiv, [Online]. Available: https://www.aptiv.com/en/solutions/advanced-safety/adas/radars. [Accessed 10 October 2023].
[62] “Fahrdynamikfläche,” Aldenhoven testing centre, [Online]. Available: https://www.aldenhoven-testing-center.de/de/strecken/fahrdynamikfl%C3%A4che.html. [Accessed 10 October 2023].
[63] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C. Tai. TransFusion: Robust lidar-camera fusion for 3D object detection with transformers. In CVPR, pages 1090–1099, 2022.
[64] N.H. Barnouti, S.S.M. Al-Dabbagh, and W.E. Matti. Face recognition: A literature review. IJAIS, 11(4):21–31, 2016.
[65] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. arXiv, 2023.
[66] 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 CVPR, pages 11621–11631, 2020.

[67] H. Cai, Z. Zhang, Z. Zhou, Z. Li, W. Ding, and J. Zhao. BEVFusion4D: Learning lidar-camera fusion under bird's eye-view via cross-modality guidance and temporal aggregation. arXiv, 2023.
[68] M. Caron, H. Touvron, I. Misra, H. J´egou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in selfsupervised vision transformers. In ICCV, pages 9650–9660, 2021.
[69] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3D object detection network for autonomous driving. In CVPR, pages 1907–1915, 2017.
[70] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss. Automatic labeling to generate training data for online lidar-based moving object segmentation. RA-L, 7(3):6107–6114, 2022.
[71] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao. FUTR3D: A unified sensor fusion framework for 3D detection. In CVPR, pages 172–181, 2023.
[72] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia. Focal sparse convolutional networks for 3D object detection. In CVPR, pages 5428–5437, 2022.
[73] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In CVPR, pages 1201–1210, 2015.
[74] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. IJCV, 88:303–338, 2010.
[75] K. Hsu, Y. Lin, and Y. Chuang. Co-attention CNNs for unsupervised object co-segmentation. In IJCAI, 2018.
[76] 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 CVPR, pages 17853–17862, 2023.
[77] T. Huang, Z. Liu, X. Chen, and X. Bai. EPNet: Enhancing point features with image semantics for 3D object detection. In ECCV, pages 35–52, 2020.
[78] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In CVPR, pages 1943–1950, 2010.
[79] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In CVPR, pages 542–549, 2012.
[80] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast encoders for object detection from point clouds. In CVPR, pages 12697–12705, 2019.
[81] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao, et al. LoGoNet: Towards accurate 3D object detection with local-to-global cross-modal fusion. In CVPR, pages 17524–17534, 2023.
[82] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia. Unifying voxel-based representation with transformer for 3D object detection. In NeurIPS, pages 18442–18455, 2022.
[83] Y. Li, A.W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q.V. Le, et al. DeepFusion: Lidarcamera deep fusion for multi-modal 3D object detection. In CVPR, pages 17182–17191, 2022.
[84] H. Liang, C. Jiang, D. Feng, X. Chen, H. Xu, X. Liang, W. Zhang, Z. Li, and L. Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3D object detection. In ICCV, pages 3293–3302, 2021.
[85] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In ECCV, pages 641–656, 2018.
[86] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang. BEVFusion: A simple and robust lidarcamera fusion framework. In NeurIPS, pages 10421–10434, 2022.
[87] Z. Lin, Z. Yang, and Y. Wang. Foreground guidance and multi-layer feature fusion for unsupervised object discovery with transformers. In WACV, pages 4043–4053, 2023.

[88] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In ICRA, pages 2774–2781, 2023.
[89] L. McInnes, J. Healy, and S. Astels. HDBSCAN: Hierarchical density based clustering. JOSS, 2(11):205, 2017.
[90] M. Najibi, J. Ji, Y. Zhou, C.R. Qi, X. Yan, S. Ettinger, and D. Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In ECCV, pages 424–443, 2022.
[91] 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. arXiv, 2023.
[92] C.R. Qi, W. Liu, C. Wu, H. Su, and L.J. Guibas. Frustum pointnets for 3D object detection from RGB-D data. In CVPR, pages 918–927, 2018.
[93] S. Shi, X.Wang, and H. Li. PointRCNN: 3D object proposal generation and detection from point cloud. In CVPR, pages 770–779, 2019.
[94] O. Sim´eoni, G. Puy, H.V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. P´erez, R. Marlet, and J. Ponce. Localizing objects with self-supervised transformers and no labels. In BMVC, 2021.
[95] V.A. Sindagi, Y. Zhou, and O. Tuzel. MVX-Net: Multimodal voxelnet for 3D object detection. In ICRA, pages 7276–7282, 2019.
[96] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In CVPR, pages 2446–2454, 2020.
[97] K. Tang, A. Joulin, L. Li, and L. Fei-Fei. Co-localization in real-world images. In CVPR, pages 1464–1471, 2014.
[98] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In CVPR, pages 2217–2224, 2011.
[99] H.V. Vo, F. Bach, M. Cho, K. Han, Y. LeCun, P. P´erez, and J. Ponce. Unsupervised image matching and object discovery as optimization. In CVPR, pages 8287–8296, 2019.
[100] H.V. Vo, P. P´erez, and J. Ponce. Toward unsupervised, multiobject discovery in large-scale image collections. In ECCV, pages 779–795, 2020.
[101] H.V. Vo, E. Sizikova, C. Schmid, P. P´erez, and J. Ponce. Large-scale unsupervised object discovery. In NeurIPS, pages 16764–16778, 2021.
[102] S. Vora, A.H. Lang, B. Helou, and O. Beijbom. PointPainting: Sequential fusion for 3D object detection. In CVPR, pages 4604–4612, 2020.
[103] C. Wang, C. Ma, M. Zhu, and X. Yang. PointAugmenting: Cross-modal augmentation for 3D object detection. In CVPR, pages 11794–11803, 2021.
[104] X. Wang, R. Girdhar, S.X. Yu, and I. Misra. Cut and learn for unsupervised object detection and instance segmentation. In CVPR, pages 3124–3134, 2023.
[105] 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 CVPR, pages 14176–14186, 2022.
[106] Y. Wang, Y. Chen, and Z. Zhang. 4D unsupervised object discovery. In NeurIPS, pages 35563–35575, 2022.
[107] Y. Wang, X. Shen, S.X. Hu, Y. Yuan, J.L. Crowley, and D. Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In CVPR, pages 14543–14553, 2022.
[108] Z. Wang and K. Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In IROS, pages 1742–1749, 2019.

[109] J. Yan, Y. Liu, J. Sun, F. Jia, S. Li, T. Wang, and X. Zhang. Cross modal transformer: Towards fast and robust 3D object detection. arXiv, 2023.
[110] Z. Yang, J. Chen, Z. Miao, W. Li, X. Zhu, and L. Zhang. DeepInteraction: 3D object detection via modality interaction. In NeurIPS, pages 1992–2005, 2022.
[111] J. Yin, D. Zhou, L. Zhang, J. Fang, C. Xu, J. Shen, and W. Wang. ProposalContrast: Unsupervised pre-training for lidar-based 3D object detection. In ECCV, pages 17–33, 2022.
[112] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3D object detection and tracking. In CVPR, pages 11784–11793, 2021.
[113] T. Yin, X. Zhou, and P. Kr¨ahenb¨uhl. Multimodal virtual point 3D detection. In NeurIPS, pages 16494–16507, 2021.
[114] J.H. Yoo, Y. Kim, J. Kim, and J.W. Choi. 3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection. In ECCV, pages 720–736, 2020.
[115] 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 CVPR, pages 1130–1140, 2022.
[116] L. Zhang, A.J. Yang, Y. Xiong, S. Casas, B. Yang, M. Ren, and R. Urtasun. Towards unsupervised object detection from lidar point clouds. In CVPR, pages 9317–9328, 2023.
[117] Y. Zhou and O. Tuzel. VoxelNet: End-to-end learning for point cloud-based 3D object detection. In CVPR, pages 4490–4499, 2018.
[118] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau and Y. Dupuis, Survey on Cooperative Perception in an Automotive Context, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14204-14223, Sept. 2022, doi: 10.1109/TITS.2022.3153815.
[119] Han, Y., Zhang, H., Li, H., Jin, Y., Lang, C., Li, Y.: Collaborative perception in autonomous driving: methods, datasets and challenges. ArXiv, abs/2301.06262 (2023).
[120] Malik S, Khan MJ, Khan MA, El-Sayed H. Collaborative Perception-The Missing Piece in Realizing Fully Autonomous Driving. *Sensors*. 2023; 23(18):7854. https://doi.org/10.3390/s23187854.
[121] Singh, G., Akrigg, S., Di Maio, M., Fontana, V., Alitappeh, R. J., Khan, S., ... & Cuzzolin, F. (2022). Road: The road event awareness dataset for autonomous driving. IEEE transactions on pattern analysis and machine intelligence, 45(1), 1036-1054.
[122] Izquierdo, R., Quintanar, A., Parra, I., Fernández-Llorca, D., & Sotelo, M. A. (2019, October). The prevention dataset: a novel benchmark for prediction of vehicles intentions. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3114-3121). IEEE.
[123] CARLA, Open-source simulator for autonomous driving research, https://carla.org/.
[124] Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., & Ma, J. (2021, September). OpenCDA: an open cooperative driving automation framework integrated with co-simulation. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) (pp. 1155-1162). IEEE.
[125] Xu, R., Xiang, H., Han, X., Xia, X., Meng, Z., Chen, C. J., ... & Ma, J. (2023). The opencda open-source ecosystem for cooperative driving automation research. IEEE Transactions on Intelligent Vehicles.
[126] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In 2022 International Conference on Robotics and Automation (ICRA), pages 2583–2589. IEEE, 2022.

[127] Li, Y., Ma, D., An, Z., Wang, Z., Zhong, Y., Chen S., Feng, C.: V2X-Sim: multi-agent collaborative perception dataset and benchmark for autonomous driving. IEEE Robotics and Automation Letters, 10914-10921 (2002).
[128] Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M. H., & Ma, J. (2022, October). V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In European conference on computer vision (pp. 107-124). Cham: Springer Nature Switzerland.
[129] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia , L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, L. Yu, 2021b. ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration. doi:10.5281/zenodo.4418161.
[130] S. Bell, C. L. Zitnick, K. Bala, R. Girshick, " Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[131] K. Li, et al, “Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping,” IEEE Robotics and Automation Letters, Vol. 6, No. 3, pp. 5167-5174, 2021.
[132] T. Shan, B. Englot et al., “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020.
[133] K. Chen, et al, “Direct LiDAR Odometry: Fast Localization with Dense Point Clouds,” IEEE Robotics and Automation Letters, Vol. 7, No. 2, pp. 2000-2007, 2022.
[134] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp.” in Robotics: Science and Systems (RSS), 2009.
[135] B. Kim et al., "Multiple relative pose graphs for robust cooperative mapping," IEEE International Conference on Robotics and Automation, pp. 3185-3192, 2010.
[136] F. Tango et al., “D.2.1: User and System Requirements for selected Use-cases,” 2023 March 2023. [Online]. Available: 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. [Accessed 26 October 2023].
[137] O. Yurduseven, T. Fromenteze, C. Decroze and V. F. Fusco, “Frequency-Diverse Computational Automotive Radar Technique for Debris Detection,” IEEE Sensors Journal, vol. 2022, pp. 13167-13177, 2020.
[138] M. Shibao and A. Kajiwara, “Road Debris Detection Using 79GHz Radar,” in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), 2019.
[139] D. Dvoryankov, D. Valuyskiy, S. Vityazev and V. Vityazev, “The Problem of Debris Detection with Automotive 77-GHz FMCW Radar,” in 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021.
[140] R. Yamada, M. Shibao and A. Kajiwara, “Radar cross section measurement of road debris in 79 GHz-band,” IEICE Communications Express, vol. 10, no. 2, pp. 81-86, 2021.
[141] M. Insider, “What Is 4D Imaging Radar?,” Aptiv, September 2021. [Online]. Available: https://www.aptiv.com/en/insights/article/what-is-4d-imaging-radar. [Accessed 6 October 2023].
[142] C. Zatout, “Point Cloud Segmentation in Python. Data clustering using scikit-learn,” [Online]. Available: https://medium.com/@chimso1994/point-cloud-segmentation-in-python-2fdbf5ea0617. [Accessed 4 October 2023].
[143] M. R. S. a. S. M. S. I. Ahmed, “"The k-means Algorithm: A Comprehensive Survey and Performance Evaluation",” Electronics , vol. 9, no. 8, 2020.
[144] K. Arvai, “K-Means Clustering in Python: A Practical Guide,” [Online]. Available: https://realpython.com/k-means-clustering-python/. [Accessed 4 October 2023].

[145] M. a. K. H.-P. a. S. J. a. X. X. a. o. Ester, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in kdd, 1996.
[146] X. Dongkuan and T. Yingjie, “A Comprehensive Survey of Clustering Algorithms.,” Annals of Data Science, vol. 2, p. 165–193, 2015.
[147] 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.
[148] Mathworks, “Track Objects in a Parking Lot Using TI mmWave Radar,” Mathworks, [Online]. Available: https://ch.mathworks.com/help/radar/ug/track-objects-parking-lot-mmwaveradar-example.html. [Accessed 5 October 2023]
[149] N. Scheiner, O. Schumann, F. Kraus, N. Appenrodt, J. Dickmann and B. Sick, “Off-the-shelf sensor vs. experimental radar - How much resolution is necessary in automotive radar classification?,” in 2020 IEEE 23rd International Conference on Information Fusion (FUSION), 2020.
[150] A. Palffy, J. Dong, J. F. P. Kooij and D. M. Gavrila, "CNN Based Road User Detection Using the 3D Radar Cube," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1263-1270, 2020.
[151] A. Palffy, J. Kooij and D. Gavrila, "Detecting darting out pedestrians with occlusion aware sensor fusion of radar and stereo camera," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 2, pp. 1459-1472, 2023.
[152] J. Lombacher, M. Hahn, J. Dickmann and C. Wöhler, "Potential of radar for static object classification using deep learning methods," in IEEE MTT-S International Conference on Microwaves for Intelligent Mobility, 2016.
[153] O. Schumann, M. Hahn, J. Dickmann and C. Wöhler, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," in Sensor Data Fusion: Trends, Solutions, Applications, 2017.
[154] A. Palffy, E. Pool, S. Baratam, J. F. P. Kooij and D. M. Gavrila, "Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4691-4968, 2022.
[155] F. Ding, A. Palffy, D. M. Gavrila and C. X. Lu, "Hidden Gems: 4D Radar Scene Flow Learning Using Cross Modal Supervision," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.
[156] F. Tango et al., “D.2.1: User and System Requirements for selected Use-cases,” 2023 March 2023. [Online]. Available: 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. [Accessed 26 October 2023].
[157] A. Ohazulike et al., “EVENTS D.2.2 Full Stack Architecture & Interfaces,” June 2023. [Online]. Available: https://www.events-project.eu/wp-content/uploads/2023/07/EVENTS_D2.2_Full-Stack-Architecture-Interfaces_v1.0.pdf. [Accessed 13 October 2023].
[158] F. Sezgin, D. Vriesman, D. Steinhauser, R. Lugner and T. Brandmeier, “Safe Autonomous Driving in Adverse Weather: Sensor Evaluation and Performance Monitoring,” in 2023 IEEE Intelligent Vehicles Symposium (IV), 2023.
[159] O. Yurduseven, T. Fromenteze, C. Decroze and V. F. Fusco, “Frequency-Diverse Computational Automotive Radar Technique for Debris Detection,” IEEE Sensors Journal, vol. 2022, pp. 13167-13177, 2020.
[160] M. Shibao and A. Kajiwara, “Road Debris Detection Using 79GHz Radar,” in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), 2019.
[161] D. Dvoryankov, D. Valuyskiy, S. Vityazev and V. Vityazev, “The Problem of Debris Detection with Automotive 77-GHz FMCW Radar,” in 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021.

[162] R. Yamada, M. Shibao and A. Kajiwara, “Radar cross section measurement of road debris in 79 GHz-band,” IEICE Communications Express, vol. 10, no. 2, pp. 81-86, 2021.
[163] “Debris Detection,” NAVTECH Radar, [Online]. Available: https://navtechradar.com/explore/debris-detection/. [Accessed October 2023].
[164] “NCSA Tools, Publications, and Data,” NHTSA, [Online]. Available: https://cdan.dot.gov/. [Accessed October 2023].
[165] “Traffic Safety Facts Annual Report Tables,” NHTSA, [Online]. Available: https://cdan.dot.gov/tsftables/tsfar.htm. [Accessed 11 October 2023].
[166] “Mobility & Transport - Latest key figures,” European Commision, [Online]. Available: https://road-safety.transport.ec.europa.eu/european-road-safety-observatory/data-and-analysis/latest-key-figures_en#collision-matrix. [Accessed 11 October 2023].
[167] “ROAD TRAFFIC FATALITIES,” European commision, [Online]. Available: https://road-safety.transport.ec.europa.eu/system/files/2022-08/road_traffic_fatalities_in_the_eu_in_2020_total.pdf. [Accessed 11 October 2023].
[168] “Lawsuits for Accidents Caused by Road Debris,” Shouse California Law Group, [Online]. Available: https://www.shouselaw.com/ca/personal-injury/car-accident/road-debris-accident-lawsuit/. [Accessed 11 October 2023].
[169] “Look Out! 10 Most Common Types of Road Debris,” Geroge Sink, P.A. Injury Lawyers, March 2013. [Online]. Available: https://www.sinklaw.com/blog/look-out-10-most-common-types-of-road-debris/. [Accessed 11 October 2023].
[170] “About AAA Foundation for Traffic Safety,” AAA Foundation for Traffic Safety, [Online]. Available: https://aaafoundation.org/about/. [Accessed 11 October 2023].
[171] 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.
[172] “How Common Is Car Damage from Road Debris?,” Injury Law PLLC Todd W.Burris Attorney, 19 November 2021. [Online]. Available: https://www.toddwburrislaw.com/how-common-is-car-damage-from-road-debris/. [Accessed 11 October 2023].
[173] “Crash Investigation Sampling System,” NHTSA, [Online]. Available: https://www.nhtsa.gov/crash-data-systems/crash-investigation-sampling-system. [Accessed 11 October 2023].
[174] B. Zhang, G. Li, Q. Zheng, X. Bai, Y. Ding and A. Khan, “Path planning for wheeled mobile robot in partially known uneven terrain,” Sensors, vol. 22, no. 14, p. 5217, 2022.
[175] T. L. M. N. D. H. V. K. B. Gabriel Günter Waibel, “How Rough Is the Path? Terrain Traversability Estimation for Local and Global Path Planning,” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2022.
[176] A. C. a. H. Hirschm¨uller, “Stereo Camera Based Navigation of Mobile Robots on Rough Terrain,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.
[177] Y. Zhou, Y. Huang and Z. Xiong, “3D Traversability Map Generation for Mobile Robots Based on Point Cloud,” in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2021.
[178] T. Guan, Z. He, R. Song, D. Manocha and L. Zhang, “TNS: Terrain Traversability Mapping and Navigation System for Autonomous Excavators,” 2022.
[179] K. Zhang, Y. Yang, M. Fu and M. Wang, “Traversability Assessment and Trajectory Planning of Unmanned Ground Vehicles with Suspension Systems on Rough Terrain,” Sensors, 2019.
[180] M. K. C. ALEXANDER AXELSSON, “A Study on the Impact of Traversability Measures on Rough Terrain Path Planning,” Master Thesis KTH, 2020.
[181] R. Counts, “Off-Road Buying Guide,” edumuns, 26 March 2020. [Online]. Available: https://www.edmunds.com/vehicles/off-road-buying-

guide/#:~:text=Generally%2C%20all%2D%20or%20four%2D,more%20should%20be%20pretty%20good.. [Accessed 12 October 2023].
[182] "What is the Difference Between Off-Road and On-Road?," off-road handbook, 27 January 2022. [Online]. Available: https://offroadhandbook.com/what-is-the-difference-between-off-road-and-on-road/?utm_content=cmp-true. [Accessed 13 October 2023].
[183] "Speed Humps vs. Speed Bumps," MaineDOT, [Online]. Available: https://web.archive.org/web/20130627171125/http://www.maine.gov/mdot/csd/mlrc/technical/shsb.htm. [Accessed 13 October 2023].
[184] J. J. Fazzalaro, "SPEED BUMPS AND SPEED HUMPS," Office of Legislative Research, 22 September 2006. [Online]. Available: https://www.cga.ct.gov/2006/rpt/2006-r-0567.htm. [Accessed 13 October 2023].
[185] "What's the Deal With Speed Bumps?," Sun Devil Auto, [Online]. Available: https://www.sundevilauto.com/whats-the-deal-with-speed-bumps/. [Accessed 13 October 2023].
[186] G. Watts, "Road humps for the control of vehicle speeds," Transport and Road Research Laboratory, Crowthorne, Berkshire, 1973.
[187] B. Zheng, Z. Hong, J. Tang, M. Han, J. Chen and X. Huang, "A Comprehensive Method to Evaluate Ride Comfort of Autonomous Vehicles under Typical Braking Scenarios: Testing, Simulation and Analysis.," Mathematics, vol. 11, no. 2, 2023.
[188] E. Lima, M. Silveira, J. A. Júnior, C. A. S. Carvalho, M. F. L. Antunes and D. M. Junqueira, "Evaluation of Vehicular Discomfort Measures Produced by Speed Bumps Using Numerical Simulation," Journal of Mechanical Engineering and Automation, vol. 5, no. 4, pp. 113-123, 2015.
[189] "Advanced Radars. Aptiv's most advanced sensors push the envelope," Aptiv, [Online]. Available: https://www.aptiv.com/en/solutions/advanced-safety/adas/radars. [Accessed 10 October 2023].
[190] "Fahrdynamikfläche," Aldenhoven testing centre, [Online]. Available: https://www.aldenhoven-testing-center.de/de/strecken/fahrdynamikfl%C3%A4che.html. [Accessed 10 October 2023].
[191] M. Insider, "What Is 4D Imaging Radar?," Aptiv, September 2021. [Online]. Available: https://www.aptiv.com/en/insights/article/what-is-4d-imaging-radar. [Accessed 6 October 2023].
[192] C. Zatout, "Point Cloud Segmentation in Python. Data clustering using scikit-learn," [Online]. Available: https://medium.com/@chimso1994/point-cloud-segmentation-in-python-2fdbf5ea0617. [Accessed 4 October 2023].
[193] M. R. S. a. S. M. S. I. Ahmed, ""The k-means Algorithm: A Comprehensive Survey and Performance Evaluation"," Electronics , vol. 9, no. 8, 2020.
[194] K. Arvai, "K-Means Clustering in Python: A Practical Guide," [Online]. Available: https://realpython.com/k-means-clustering-python/. [Accessed 4 October 2023].
[195] M. a. K. H.-P. a. S. J. a. X. X. a. o. Ester, "A density-based algorithm for discovering clusters in large spatial databases with noise," in kdd, 1996.
[196] X. Dongkuan and T. Yingjie, "A Comprehensive Survey of Clustering Algorithms.," Annals of Data Science, vol. 2, p. 165–193, 2015.
[197] 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.
[198] Mathworks, "Track Objects in a Parking Lot Using TI mmWave Radar," Mathworks, [Online]. Available: https://ch.mathworks.com/help/radar/ug/track-objects-parking-lot-mmwaveradar-example.html. [Accessed 5 October 2023].
[199] N. Scheiner, O. Schumann, F. Kraus, N. Appenrodt, J. Dickmann and B. Sick, "Off-the-shelf sensor vs. experimental radar - How much resolution is necessary in automotive

radar classification?," in 2020 IEEE 23rd International Conference on Information Fusion (FUSION), 2020.
[200] D.-H. Paek, S.-H. Kong and K. T. Wijaya, "K-Radar: 4D Radar Object Detection for Autonomous Driving in Various Weather Conditions," in Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.
[201] K. Tyagi, S. Zhang, Y. Zhang, J. Kirkwood, S. Song and N. Manukian, "Machine Learning Based Early Debris Detection Using Automotive Low Level Radar Data," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023.
[202] 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.
[203] I. Benmahammed, "How to add confidence to your Machine Learning models," TotalEnergies Digital Factory, April 2022. [Online]. Available: https://medium.com/totalenergies-digital-factory/how-to-add-confidence-to-your-machine-learning-models-b1228217858e. [Accessed October 2023].
[204] S. Cloud, "How to Get a Confidence Measure for Each Prediction in a Machine Learning Model Python," June 2023. [Online]. Available: https://saturncloud.io/blog/how-to-get-a-confidence-measure-for-each-prediction-in-a-machine-learning-model-python/. [Accessed October 2023].
[205] "Lecture 15: Gaussian Processes.," Cornell Bowers CIS, [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html. [Accessed 12 October 2023].
[206] "Tutorial: Basic Statistics in Python — Probability," DATAQUEST, 18 July 2018. [Online]. Available: https://www.dataquest.io/blog/basic-statistics-in-python-probability/. [Accessed 13 October 2023].
[207] l. Musralina, T. Zwick and M. Harter, "Height Estimation Methods for Object Detection in Automotive Radar Applications," in Proceedings of the 8th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'22), 2022.
[208] A. Laribi, M. Hahn, J. Dickmann and C. Waldschmidt, "A new height-estimation method using FMCW radar Doppler beam sharpening," in 2017 25th European Signal Processing Conference (EUSIPCO), 2017.
[209] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother and R. Mester, "Lost and Found: detecting small road hazards for self-driving vehicles," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
[210] T. E. Choe, J. Wu, X. Lin, K. Kwon and M. Park, "HazardNet: Road Debris Detection by Augmentation of Synthetic Models," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023.
[211] S. Melo, E. Marchetti, S. Cassidy, E. Hoare, A. Bogoni, M. Gashinova and M. Cherniakov, "24 GHz Interferometric Radar for Road Hump Detections in Front of a Vehicle," in 2018 19th International Radar Symposium (IRS), 2018.
[212] EVENTS partners, "D.2.1: User and System Requirements for selected Use-cases," 2023.
[213] EVENTS partners, "D.2.2 Full Stack Architecture & Interfaces," 2023.
[214] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang y H. Chen, «A Survey on Trajectory-Prediction Methods for Autonomous Driving,» IEEE Transactions on Intelligent Vehicles, 2022.
[215] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey and D. Ramanan, "Argoverse: 3d tracking and forecasting with rich maps," in IEEE/CVF conference on computer vision and pattern recognition, 2019.
[216] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle and M. Tomizuka, "INTERACTION Dataset: An INTERnational,

Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps".
[217] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in Conference on Robot Learning, 2021.
[218] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou and others, "Large scale interactive motion forecasting for autonomous driving: The Waymo open motion dataset," in IEEE/CVF International Conference on Computer Vision, 2021.
[219] 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 IEEE/CVF conference on computer vision and pattern recognition, 2020.
[220] A. Malinin, N. Band, G. Chesnokov, Y. Gal, M. J. Gales, A. Noskov, A. Ploskonosov, L. Prokhorenkova, I. Provilkov, V. Raina and others, "Shifts: A dataset of real distributional shift across multiple large-scale tasks," 2021.
[221] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes and others, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 2021.
[222] Z. Zhou, L. Ye, J. Wang, K. Wu and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[223] W. J. R. X. W. Z. J. M. S. L. Y. L. Xinyu Cai, «Analyzing Infrastructure LiDAR Placement with Realistic LiDAR Simulation Library,» de IEEE International Conference on Robotics and Automation (ICRA), 2022.
[224] W. Dongkai and S. Zhang, „Contextual instance decoupling for robust multi-person pose estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[225] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union," June 2019.
[226] H. W. Kuhn, "The Hungarian method for the assignment problem," Naval research logistics quarterly, vol. 2, no. 1-2, pp. 83–97, 1955.
[227] A. Holzbock, A. Tsaregorodtsev and V. Belagiannis, Pedestrian Environment Model for Automated Driving, arXiv preprint arXiv:2308.09080, 2023.
[228] C. Holger and e. al., „nuScenes: A multimodal dataset for autonomous driving," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020.
[229] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, „CARLA: An open urban driving simulator," in Proceedings of the 1st Annual Conference on Robot Learning, 2017.
[230] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern and K. Dietmayer, „A random finite set approach for dynamic occupancy grid maps with real-time application," in The International Journal of Robotics Research, vol. 37, no. 8, pp. 841–866, 2018.
[231] C. Buerkle, F. Oboril, J. Jarquin and K. Scholl, „Efficient dynamic occupancy grid mapping using non-uniform cell representation," in IEEE IV 2020, Proceedings 1629–1634, 2020.
[232] C. Wellhausen, J. Clemens and K. Schill, „Efficient grid map data structures for autonomous driving in large-scale environments," in IEEE ITSC 2021, Proceedings, 2855–2862, 2021.
[233] T. Wodtko, T. Griebel and M. Buchholz, „Adaptive Patched Grid Mapping," in arXiv preprint arXiv:2308.03416, 2023.
[234] C. Hermann, M. Herrmann, T. Griebel, M. Buchholz and K. Dietmayer, „The Fast Product Multi-Sensor Labeled Multi-Bernoulli Filter," in 2023 26th International Conference on

Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION52260.2023.10224189, Charleston, SC, USA, 2023.
[235] T. Griebel, J. Müller, M. Buchholz and K. Dietmayer, „Kalman Filter Meets Subjective Logic: A Self-Assessing Kalman Filter Using Subjective Logic," in 2020 IEEE 23rd International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION45008.2020.9190520, Rustenburg, South Africa, 2020.
[236] T. Griebel and e. al., „Self-Assessment for Single-Object Tracking in Clutter Using Subjective Logic," in 2022 25th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION49751.2022.9841294, Linköping, Sweden, 2022.
[237] T. Griebel, J. Heinzler, M. Buchholz and K. Dietmayer, „Online Performance Assessment of Multi-Sensor Kalman Filters Based on Subjective Logic," in 2023 26th International Conference on Information Fusion (FUSION), pp. 1-8, doi: 10.23919/FUSION52260.2023.10224188, Charleston, SC, USA, 2023.
[238] Dorri, A., Kanhere, S.S., & Jurdak, R. (2018). Multi-Agent Systems: A Survey. IEEE Access, 6, 28573-28593.
[239] Dorigo M, Theraulaz G, Trianni V. Swarm robotics: Past, present, and future [point of view]. Proc IEEE. 2021;109(7):1152–1165.
[240] Allidina, T.; Deka, L.; Paluszczyszyn, D.; Elizondo, D. Selecting Non-Line of Sight Critical Scenarios for Connected Autonomous Vehicle Testing. Software 2022, 1, 244–264. https://doi.org/10.3390/software1030011.
[241] Bai, Z., Wu, G., Barth, M. J., Liu, Y., Sisbot, E. A., Oguchi, K., & Huang, Z. (2022). A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation. arXiv preprint arXiv:2208.10590.
[242] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.
[243] Godoy, J.; Jiménez, V.; Artuñedo, A.; Villagra, J. A Grid-Based Framework for Collective Perception in Autonomous Vehicles. Sensors 2021, 21, 744. https://doi.org/10.3390/s21030744.
[244] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea (South), 2015, pp. 1074-1081, doi: 10.1109/IVS.2015.7225827.
[245] Nuss D, Reuter S, Thom M, et al. A random finite set approach for dynamic occupancy grid maps with real-time application. The International Journal of Robotics Research. 2018;37(8):841-866. doi:10.1177/0278364918775523.
[246] [5] R. Danescu, F. Oniga and S. Nedevschi, "Modeling and Tracking the Driving Environment With a Particle-Based Occupancy Grid," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1331-1342, Dec. 2011, doi: 10.1109/TITS.2011.2158097.
[247] Jen-Yeu Chen and Jianghai Hu, "Probabilistic Map Building by Coordinated Mobile Sensors," 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 2006, pp. 807-812, doi: 10.1109/ICNSC.2006.1673250.
[248] Coué C, Pradalier C, Laugier C, Fraichard T, Bessière P. Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. The International Journal of Robotics Research. 2006;25(1):19-30. doi:10.1177/0278364906061158.
[249] A. Nègre, L. Rummelhard and C. Laugier, "Hybrid sampling Bayesian Occupancy Filter," 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 2014, pp. 1307-1312, doi: 10.1109/IVS.2014.6856554.
[250] T. Gindele, S. Brechtel, J. Schroder and R. Dillmann, "Bayesian Occupancy grid Filter for dynamic environments using prior map knowledge," 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 2009, pp. 669-676, doi: 10.1109/IVS.2009.5164357.

[251] Tay, M.K. et al. (2008). The Bayesian Occupation Filter. In: Bessière, P., Laugier, C., Siegwart, R. (eds) Probabilistic Reasoning and Decision Making in Sensory-Motor Systems. Springer Tracts in Advanced Robotics, vol 46. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-79007-5_4.

[252] Li Y, Ma D, An Z, et al (2022) V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. IEEE Robotics and Automation Letters 7(4):10914–10921.

[253] X. Cai, W. Jiang, R. Xu, W. Zhao, J. Ma, S. Liu και Y. Li, “Analyzing Infrastructure LiDAR Placement with Realistic LiDAR Simulation Library,“ IEEE International Conference on Robotics and Automation (ICRA), 2022.

[254] Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." Conference on robot learning. PMLR, 2017.

[255] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.