

LiDAR-based Weather Detection: Automotive LiDAR Sensors in Adverse Weather Conditions

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)**

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Robin Klaus Heinzler

geb. in Karlsruhe

Tag der mündlichen Prüfung:

10.03.2022

Hauptreferent:

Prof. Dr. rer. nat. Wilhelm Stork

Korreferent:

Prof. Dr. rer. nat. Cornelius Neumann

Abstract

Technological improvements are increasing the degree of automation of vehicles. The natural move is to support the driver where they most desire: adverse weather. Weather impacts all the sensors used to perceive the surroundings, so it is crucial to factor in and mitigate these effects. This dissertation focuses on the emerging technology of automotive **L**ight **D**etection and **R**anging (**LiDAR**) sensors and contributes to the development of autonomous vehicles capable of operating under different weather conditions.

The foundation is the first **LiDAR** point cloud dataset with an emphasis on adverse weather conditions, containing pointwise ground truth labels and being recorded under well controlled weather conditions. Different weather conditions generated in controlled environments enable the aforementioned approaches to be evaluated and provide valuable information for automated and autonomous driving. This dataset is enriched by a novel weather augmentation system for generating realistic weather effects.

A unique approach to classifying weather conditions and the first CNN-based de-noising algorithm have been developed, resulting respectively in an accurate prediction of weather conditions and improved point cloud quality. Compared with conventional geometric approaches, our algorithm is capable of generalizing the underlying noise pattern, thereby allowing near-range clutter caused by fog or rain clouds to be distinguished from solid objects like pedestrians, cyclists or vehicles.

In conclusion, this thesis will make a significant contribution toward the safety of autonomous vehicles in the future through not only the efficient filtering of adverse weather conditions wherever necessary but also the accurate classification of weather conditions.

Zusammenfassung

Technologische Verbesserungen erhöhen den Automatisierungsgrad von Fahrzeugen. Der natürliche Schritt ist dabei, den Fahrer dort zu unterstützen, wo er es am meisten wünscht: bei schlechtem Wetter. Das Wetter beeinflusst alle Sensoren, die zur Wahrnehmung der Umgebung verwendet werden, daher ist es entscheidend, diese Effekte zu berücksichtigen und abzuschwächen. Die vorliegende Dissertation konzentriert sich auf die gerade entstehende Technologie der automobilen LiDAR-Sensoren und trägt zur Entwicklung von autonomen Fahrzeugen bei, die in der Lage sind, unter verschiedenen Wetterbedingungen zu fahren.

Die Grundlage ist der erste LiDAR-Punktwolken-Datensatz mit dem Schwerpunkt auf schlechte Wetterbedingungen, welcher punktweise annotierte Wetterinformationen enthält, während er unter kontrollierten Wetterbedingungen aufgezeichnet wurde. Dieser Datensatz wird durch eine neuartige Wetter-Augmentation erweitert, um realistische Wettereffekte erzeugen zu können.

Ein neuartiger Ansatz zur Klassifizierung des Wetterzustands und der erste CNN-basierte Entrauschungsalgorithmus werden entwickelt. Das Ergebnis ist eine genaue Vorhersage des Wetterstatus und eine Verbesserung der Punktwolkenqualität.

Kontrollierte Umgebungen unter verschiedenen Wetterbedingungen ermöglichen die Evaluierung der oben genannten Ansätze und liefern wertvolle Informationen für das automatisierte und autonome Fahren.

Acknowledgments

This dissertation would not have been possible without the extensive support of and discussions with many people over the past few years. In particular, I would like to thank the following people for their support, supervision and guidance: Prof. Dr. rer. nat. Wilhelm Stork, my supervising professor, and Dr. Philipp Schindler. My contribution to the scientific community and the success of my dissertation would not have been possible without their support and the numerous valuable discussions I engaged in with them.

Furthermore, I would like to thank Prof. Dr. rer. nat. Cornelius Neumann of the Light Technology Institute at the **K**arlsruhe **I**nstitute of **T**echnology (**KIT**) for his kind willingness to be a second reviewer.

Thanks also go to my colleagues at Mercedes Benz in the **LiDAR** Team for numerous and valuable discussions as well as their interest in my work. I would especially like to thank my supervisor Dr. Philipp Schindler for his dedication, support and corrections. In addition, special thanks go to Prof. Dr. Markus Enzweiler and Dr. Florian Piewak for their proofreading and feedback, which has improved the content and quality of this work.

Many thanks also to the student contributors without whom the content would be different.

Additionally, I would like to thank my family for their support in various ways throughout my education and the opportunity to pursue my interests in reaching my objectives. My deep thanks to Maren, for her understanding, patience, support and willingness to contribute in proofreading this work.

Contents

Abstract	i
Zusammenfassung	iii
Acknowledgments	v
Abbreviations and Symbols	xi
1 Introduction	1
1.1 Purpose	2
1.1.1 Automated and Autonomous Driving	2
1.1.2 Perception for Autonomous Driving	3
1.1.3 Perception in Adverse Weather Conditions	4
1.2 Scientific Contribution	5
1.3 Structure of the Work	6
2 Background	9
2.1 Light Detection and Ranging	9
2.1.1 Measurement Principle	11
2.1.2 Data Representation	13
2.2 Scattering and Absorption	17
2.2.1 Propagation in the Atmosphere	17
2.2.2 Electromagnetic Scattering	18
2.3 Weather	21
2.3.1 Fog	21
2.3.2 Rain	26
2.4 Weather Simulation in Controlled Environments	28
2.4.1 Artificial Rainfall	30

2.4.2	Artificial Fog	31
2.5	Machine Learning in Computer Vision	33
2.5.1	Feature Extraction	34
2.5.2	Machine Learning Methods	35
2.5.3	k Nearest Neighbor	35
2.5.4	Support Vector Machine	36
2.5.5	Artificial Neural Networks	38
2.5.6	Convolutional Neural Networks	43
2.5.7	Evaluation Metrics	44
3	The State of the Art	47
3.1	Weather Condition Monitoring	47
3.1.1	Measurement of Meteorological Optical Range	48
3.1.2	Measurement of Precipitation	50
3.1.3	Qualitative Measurement in Non-Stationary Systems	50
3.1.4	Qualitative Measurement with Non-Dedicated Sensors	51
3.2	Influence of Weather on LiDAR Sensors	53
3.2.1	Influence of Fog on LiDAR Sensors	53
3.2.2	Influence of Rain on LiDAR Sensors	54
3.3	Point Cloud De-Noising	55
3.3.1	Dense Point Cloud De-Noising	56
3.3.2	Sparse Point Cloud De-Noising	57
3.4	Datasets	58
3.5	Discussion of the Current State of the Art	58
3.5.1	Weather Impact and Classification	59
3.5.2	Weather De-Noising	60
3.5.3	Dataset	60
4	Concept and Method	61
4.1	Methods	61
4.1.1	Methods for Quantifying the Influence of Weather	62
4.1.2	Methods for Weather Detection by LiDAR Sensors	65
4.1.3	Methods for Point Cloud De-Noising	67
4.2	Data Acquisition	68
4.2.1	Requirements Regarding Datasets	69

4.2.2	LiDAR Sensors Used	71
4.2.3	Recorded Datasets	73
4.2.4	Data Split	77
4.3	Ground Truth Labeling and Data Augmentation	78
4.3.1	Framewise Labeling	78
4.3.2	Pointwise Labeling	79
4.3.3	Data Augmentation	81
5	LiDAR in Adverse Weather Conditions	85
5.1	Influence of Weather on Lidar Sensors	85
5.1.1	Influence of Weather on Point Clouds	85
5.1.2	Influence of Weather on Object Perception	87
5.1.3	Influence of Weather on Extracted Features	96
5.1.4	Influence of Weather on Image Representation	100
5.2	Weather Classification by LiDAR Sensors	103
5.2.1	Weather Classification by Manually Extracted Features	103
5.2.2	Weather Classification by CNN	104
5.3	Pointwise Weather Segmentation	114
5.3.1	Experiments	114
5.3.2	Results with Static Chamber Data	116
5.3.3	Results with Dynamic Chamber Data	117
5.3.4	Results with Dynamic Road Data	121
5.3.5	Visibility Estimation	121
6	Conclusions and Outlook	125
6.1	Summary and Conclusion	125
6.2	Outlook	126
A	Appendix	129
	Bibliography	131
	Personal Publications	149
	Journal article	149
	Conference proceedings	149
	Patents	149

Supervised Student Work 151

Abbreviations and Symbols

Abbreviations

ABS	Anti-lock B raking S ystem
ADAS	Advanced D river A ssistance S ystems
AEB	Autonomous E mergency B raking
AMS	American M eteorological S ociety
ANN	artificial n eural n etwork
CIE	International C ommission on I llumination
CNN	convolutional n eural n etwork
DROR	d ynamic r adius o utlier r emoval
ELU	Exponential Linear Unit
EPW	echo p ulse w idth
ESC	Electronic S tability C ontrol
EVT	E uro V ehicle T arget
FFN	f eed f orward n etwork
FMCW	f requency m odulated c ontinuous w ave
FoV	F ield o f V iew
FN	f alse n egative
FP	f alse p ositive

FPR	true positive rate
ICA	independent component analysis
IoU	intersection over union
mIoU	mean intersection over union
IVC	International Visibility Code
KIT	Karlsruhe Institute of Technology
kNN	k nearest neighbors
Laser	Light Amplification by Stimulated Emission of Radiation
LDA	linear discriminant analysis
LiDAR	Light Detection and Ranging
LUH	Leibniz University Hanover
MOR	meteorological optical range
NCAP	New Car Assessment Program
PCA	principle component analysis
PSD	particle size distribution
PWRI	Public Works Research Institute
RaDAR	radio detection and ranging
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROI	Region of Interest
ROR	radius outlier removal
SAE	Society of Automotive Engineers
SNR	signal to noise ratio

SOR	statistical o utlier r emoval
SVM	S upport V ector M achine
TN	t ru e n egative
ToF	t ime o f f light
TPR	f alse p ositive r ate
TP	t ru e p ositive
VSR	V irginia S mart R oads
VRU	vulnerable r oad u ser
2D	2 - D imensional
3D	3 - D imensional

Latin Symbols and Variables

General Definitions for Variables

x	Scalar (italic)
c	Constant scalar (lowercase, non-italic)
\mathbf{v}	Vector of arbitrary size (bold, italic)
\mathbf{M}	Matrix of arbitrary size (bold, uppercase, non-italic)
$\mathbf{M}_{i,j}$	Matrix item at position (i,j) (bold, uppercase, non-italic)

Weather, Scattering and Absorption Variables

V	Visibility
β	Specific attenuation
R	Rainfall rate
T	Visual threshold in percentage
P_{TH}	Power threshold indicating where the backscatter light results during detection by the receiver
P_{Rx}	Power of the backscatters light
C_{ext}	Extinction cross-section
$I(z)$	Incoming energy flux
$T(z)$	Fraction of energy transmitted
Q_{ext}	Extinction efficiency: $c_{ext}/\pi r^2$
Q_{sca}	Sum of scattering efficiency
Q_{abs}	Sum of absorption efficiency
C_x	Apparent luminance of a black object

C_0	Apparent luminance of the background
$LN(\mu, \sigma^2)$	Logarithmic normal function with scalar arguments

LiDAR Point Clouds

Point Cloud

Δt	Time difference between t_0 and t_1 : $\Delta t = t_1 - t_0$
c	Speed of light (constant)
λ	Wavelength
f	Frequency
$n(r)$	Number of particles per unit volume and radius increment
t	Timestamp of measurement
k	Discrete time stamp of measurement
d	Measured distance of the sensor
r	Measured distance of the sensor
ζ	Measured intensity of the sensor
e	Measured echo or return of the sensor
epw	Measured echo pulse width of the sensor
θ	Elevation angle
φ	Azimuth angle
P	Point cloud or point list of one LiDAR frame (matrix)
D	Point cloud in image representation containing the distance measurements (matrix)

\mathbf{p}	Single point of a LiDAR point cloud containing the Cartesian coordinates and additional point cloud attributes (vector)
p_x	x-value of the Cartesian coordinates of a point vector \mathbf{p}
p_y	y-value of the Cartesian coordinates of a point vector \mathbf{p}
p_z	z-value of the Cartesian coordinates of a point vector \mathbf{p}
σ	Variance
μ	Mean

Point Cloud Features

PD	Point density metric
O	Object label
N	Number of points

Machine Learning Variables

b	Bias of a neuron
w	Weight of a neuron (scalar)
\mathbf{w}	Weight vector
x	Input variable (scalar)
\mathbf{x}	Input vector
y	Output variable (scalar)
\mathbf{y}	Output vector
$f_{act}(\cdot)$	Activation function
C_i	Class with index i

$P(C_i \mathbf{x})$	Conditional probability for input vector \mathbf{x} and class C_i
$E(\mathbf{w})$	Error function

1 Introduction

In 2018, annual traffic deaths reached 1.35 million people worldwide [1] - that is enough people to fill 1,500 of the world's biggest aircraft with people every year or six every single day ¹. This comparison with air transportation illustrates the extremely high number of road-related fatalities, but still the car remains our primary means of everyday transportation. In contrast, air travel would be inconceivable were six of the biggest planes in the world to crash every day without survivors.

The most frequent victims of traffic accidents are pedestrians, cyclists and motorcyclists - vulnerable road users (VRUs) - accounting for more than half of fatalities. For people in the age group 5-29, traffic accidents are the leading cause of death [1]. The main cause of accidents resulting in personal injury is driver error. This statistic is based on data from Germany, where 65% of more than 5.4 million accidents resulting in personal injury over the last ten years were caused by driver error [3] ².

In a survey encompassing more than 1,000 respondents, 68% stated that driving in extreme weather conditions caused the most difficulties, followed by driving in the dark (37%) [4]. The full results of the survey can be seen in Figure 1.1. To the best of our knowledge, no dedicated driver assistance systems designed to support drivers during adverse weather conditions - besides basic systems like the Anti-lock Braking System (ABS) or Electronic Stability Control (ESC) - are currently available on the market. Future Advanced Driver Assistance Systems (ADAS) should be designed to support drivers where they need this most, which means that adverse weather conditions needs to be explicitly considered during design and development. The environment perception capability of these

¹ According to the European Aviation Safety Agency, the Airbus A380 has a capacity of 868 passengers [2], making it the largest passenger airplane as of 2021.

² Figures are rounded. The exact figures are 5,488,223 accidents, with 3,617,522 caused by driver error in the period of 2010 to 2019 [3]

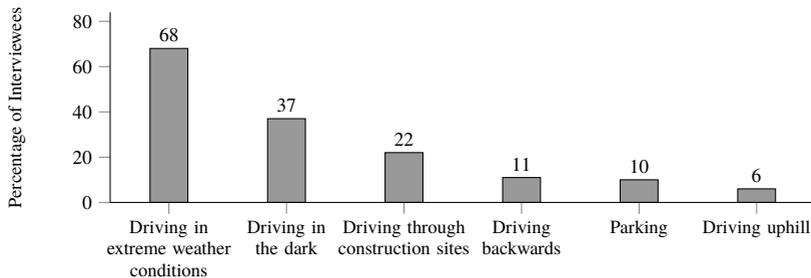


Figure 1.1: The biggest difficulties that German drivers experienced on the road in 2015. In a survey, 1,000 participants responded to the question, "Which of the following situations cause(s) you the most difficulties while driving". Only three answers were allowed, and the participants were driver's license holders who had driven at least once in the last 12 months [4].

systems is typically based on cameras, **radio detection and ranging (RaDAR)**, ultrasonic systems and **LiDAR** sensors. While the first three technologies are well-established in the industry, the latter has only recently started to emerge. This work is therefore dedicated to the impact, detection and mitigation of adverse weather conditions on **LiDAR** sensors.

1.1 Purpose

1.1.1 Automated and Autonomous Driving

For on-road automated driving systems, the **Society of Automotive Engineers (SAE)** has defined the five levels of automation [5]. At levels 1 and 2, the driver is always in charge of the vehicle, although the system is capable of performing one or two driving tasks (e.g. longitudinal and lateral control). At level 3, the system assumes partial responsibility for driving, for example, the system is capable of taking over control on specific highways until an unusual traffic situation occurs (e.g. an accident), the highway ends or the driver wants to leave the highway. At Level 4, the system assumes full responsibility in certain,

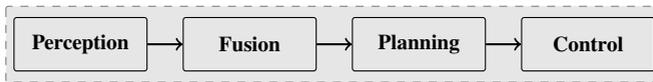


Figure 1.2: Chronological sequence of interaction among the individual subsystems for autonomous driving. The environment is first perceived by a number of sensors, and the resulting data is then fused to map the environment. This is used to calculate the planning path. Finally, the control systems for the lateral and longitudinal actuators ensure that the selected trajectory is followed.

defined use cases (e.g. during daytime in a defined urban district). At level 5, the system assumes total control in all situations and without restriction.

To realize these complex driving tasks in a technical system, various subsystems with subtasks are presented [6, 7]. In Figure 1.2, the tasks are split into four major subcomponents [6, 7]. Environment perception is the first instance providing reliable information about the surroundings for further processing steps such as fusion, planning and control. The environment is typically perceived using multiple sensors. Missing or incorrect information in one of the subsystems cannot - or can only partially - be compensated for or identified by downstream systems. Since the extent to which environment perception is successful can impact downstream systems, it plays a key role in the availability and performance of the entire autonomous system.

1.1.2 Perception for Autonomous Driving

The most challenging aspect of achieving good environmental perception capability is developing a system that functions not under defined conditions only or only sometimes, but one that functions all the time. This could be interpreted as the shift from a level 4 to a level 5 system, whereby the autonomous system is capable of operating at all times under all conditions. But since a level 3 system needs to be capable of recognizing its own boundaries independently before returning control to the driver when the defined, level 3 conditions are not fulfilled, the step from level 2 to level 3 is significant. The capability to follow the driver ahead on an empty highway in ideal weather conditions was

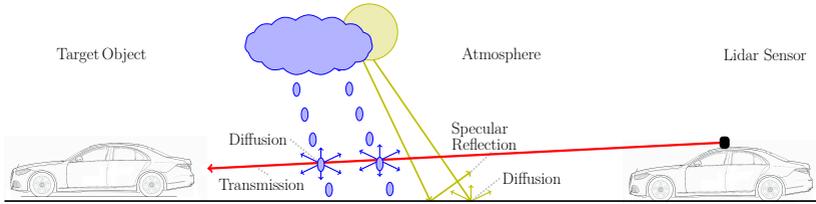


Figure 1.3: Schematic overview of influences on the light emitted from LiDAR sensors. The transmitted and received signal is influenced by the sensor and its components, by the atmosphere and by the target object. This diagram is based on [16] and was expanded accordingly.

developed a decade ago³; the challenge now, however, is to develop a system capable of properly perceiving and responding to unusual situations that even humans rarely experience. Multimodal sensor setups are therefore typically used to ensure robust and redundant environment perception. LiDAR sensors are increasingly being developed and used in the automotive field [8–15]. By scanning their surroundings with light in the near-infrared range, LiDAR sensors create a 3-Dimensional (3D) representation of the environment, which completes the perception of established sensors such as cameras, radar or ultrasonic and enables redundancies.

1.1.3 Perception in Adverse Weather Conditions

A range of environmental conditions and traffic situations need to be addressed in order to increase the degree of automation and system availability. To ensure the safety of a system, it is vital to understand every conceivable environmental condition and recognize how they affect the system response. For level 3 systems, for example, all the influences of different weather conditions on the raw sensor data and underlying algorithms need to be addressed so that the system limits can be identified. Figure 1.3 shows the interaction between LiDAR sensors and the environment, especially under different weather conditions. The diagram has been simplified to show one single laser pulse and the weather influences based on water droplets and sunlight. The main factors influencing

³ "Empty" here refers to the absence of objects of any kind

the performance of the **LiDAR** sensors are the sensor itself, the target objects in the environment and the atmosphere between these two objects. The atmosphere includes all influencing factors that affect the emitted and reflected light waves between the sensor and the object (e.g. droplets of rain).

During autonomous driving, therefore, when sensors are constantly faced with unique and varying atmospheric conditions, research into environmental influences on sensor performance is absolutely vital and essential to the development of autonomous systems capable of operating in adverse weather conditions - i.e. the driving conditions that drivers reported as being the most challenging and problematic [4].

1.2 Scientific Contribution

This work is intended as a contribution to the development of autonomous vehicles capable of operating under a range of weather conditions. The main focus is on **LiDAR** sensor perception algorithms and concepts for rain and fog. This work addresses the following scientific questions:

- To what extent are **LiDAR** sensors affected by adverse weather conditions, especially dense fog and heavy rainfall?
- Is it possible to develop an algorithm concept that can determine the weather conditions only on the basis of the raw data from the **LiDAR** sensor?

In addition to an analysis of the influence of weather and the classification of weather condition, this dissertation also discusses the following questions:

- Is it possible to develop an algorithm that segments the **LiDAR** points according to whether a point is caused by unfavorable weather or by a solid object?
- Is it possible to estimate the maximum sensor viewing range using only the point cloud data?

To answer these questions, a dataset of **LiDAR** point clouds captured during different weather conditions is required. Since no public dataset of this kind

currently exists, appropriate datasets were created while this thesis was being prepared. Since the dataset was recorded both in a dedicated climate chamber and on the road, it contains highly accurate weather information derived from reproducible and real-world scenarios. The dataset therefore enables the following, which are part of this work:

1. The first **LiDAR** point cloud dataset with an emphasis on adverse weather conditions, containing pointwise ground truth labels and being recorded under controlled weather conditions. The dataset is publicly available.
2. The detailed analysis of the influence of weather on **LiDAR** sensors under well-known ground truth weather conditions provides valuable information for automated and autonomous driving.
3. A novel approach classifying the weather conditions based on a **LiDAR** point cloud for controlled and uncontrolled environments is presented.
4. The first **LiDAR** point cloud de-noising approach based on convolutional neural networks (**CNNs**) resulting in significant performance gains over previous state-of-the-art geometric approaches while being highly efficient in terms of execution time.
5. Weather augmentation for enriching **LiDAR** point clouds with realistic weather effects.
6. Evaluation of de-noising algorithms in controlled environments under different weather conditions based on a point-level ground truth.

1.3 Structure of the Work

To help readers navigate through the individual topics, the structure of this work is outlined.

Chapter 1 provides an introduction to and describes the purpose of this work.

Chapter 2 outlines the technical background, including the fundamentals of **LiDAR** sensors, weather, scattering and absorption, machine learning and evaluation metrics.

Chapter 3 describes the current state of the art of **LiDAR** sensors in adverse weather conditions, with an focus on weather influences, weather classification and de-noising.

Chapter 4 details the concept and methods behind this work.

Chapter 5 describes and analyzes the influence of weather on **LiDAR** sensors, weather classification by **LiDAR** sensors and the pointwise weather semantic including the de-noising of point clouds.

Chapter 6 summarizes the results and looks at the prospects for future applications in this research field.

2 Background

This chapter provides a comprehensive description of the underlying principles. [Section 2.1](#) provides some technical background to [LiDAR](#) sensors. [Section 2.2](#) outlines the fundamentals regarding the interaction of small particles like water droplets with light pulses. [Section 2.3](#) covers research into weather, including the definition, characteristics and simulation of rain and fog. [Section 2.5](#), the concluding section, covers the fundamentals of machine learning with an emphasis on computer vision applications.

2.1 Light Detection and Ranging

This section contains technical background information on state-of-the-art [LiDAR](#) sensors, which is important for understanding the context of this work. The term [LiDAR](#) itself already explains the basic functionality of a [LiDAR](#) sensor. The sensor emits light in the near-infrared range from $800\text{ nm} - 2.5\text{ }\mu\text{m}$, which is reflected by a target in the environment and then detected by the sensor's receiver. The distance d to the target can then be calculated, as shown in the diagram in [Figure 2.1](#).

The essential functions of a [LiDAR](#) sensor can be divided into five major components: (1) *illumination*, (2) *wavelength*, (3) *steering*, (4) *receiver* and (5) *distance measurement*. Various technologies are available for each component, some of which can be seen in [Figure 2.2](#). By combining these technologies,

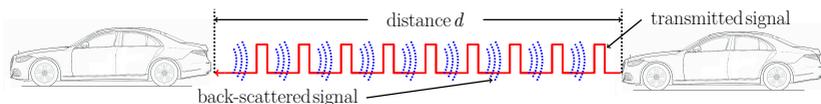


Figure 2.1: Principle measurement method of [LiDAR](#) sensors.

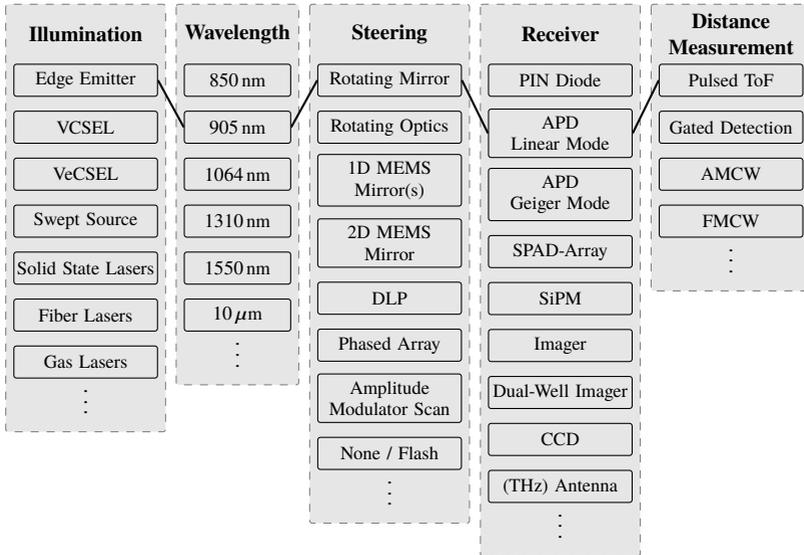


Figure 2.2: Principle components and technologies of **LiDAR** sensors. One possible combination of technologies is indicated as used in the 'Scala' **LiDAR** sensor which is used in this work.

developers can create many different **LiDAR** sensor types. This is also reflected in numerous different sensor manufacturers currently active on the market (e.g. Aeva Inc. [8], Baraja Pty Ltd. [9], Blickfeld GmbH [10], Hesai Technology [11], Innoviz Technologies Ltd. [12], Valeo [13], Velodyne Lidar Inc. [14], Waymo LLC [15]). One possible combination of subsystems is given in [Figure 2.2](#), this is the combination of used for the 'Scala' sensor. The Velodyne **LiDAR** sensors are based on similar technology combinations, but use rotating optics instead of a rotating mirror. Unlike with cameras or **RaDAR**, a system design that provides the optimal trade-off between cost, performance and manufacturability has yet to emerge (as at 2021). For the sake of simplicity, only sensor designs which used in this work are explained in more detail. For a complete overview of **LiDAR** technologies currently available and for further reading, see [17, p. 1-18], [18, p. 405-430] and [19].

2.1.1 Measurement Principle

The most commonly used method for range measurement is pulsed **time of flight (ToF)**, although a number of methods - such as **frequency modulated continuous wave (FMCW)** - exist for deriving distance based on the transmitted and received signal. For the purpose of this thesis, only **ToF** is explained in detail, because the sensors utilized are based on this principle. For other range measurement principles, [19] is recommended.

Distance Measurement

When the speed of light c and the time $\Delta t = t_1 - t_0$ between transmission t_0 and reception t_1 is known, the distance d to the object is defined by

$$d = \frac{\Delta t \cdot c}{2}. \quad (2.1)$$

Consequently, the **ToF** method measures the distance directly and the resolution in distance results from the resolution in time of the receiver, which is typically in the nanosecond range. The distance resolution is thus in the centimeter range. [Figure 2.3](#) shows a diagram of the transmitted pulse and target reflection. Whether an object is detected depends on whether a reflected light pulse can be distinguished from the background noise, caused by other light sources, above all natural sunlight and the sensor components themselves. The maximum detection range of a sensor therefore depends on the **signal to noise ratio (SNR)**.

Furthermore, if the receiver component is capable of detecting multiple values above the threshold P_{th} , the system can provide multiple distance measurements from one receiver component. This multi-return capability provides valuable information whenever the emitted light pulse is only partially backscattered by an object and then later backscattered again by another object. This often occurs in the vicinity of vegetation, for example, where part of the light is reflected by a leaf and the other part hits the trunk of a tree. This behavior generally occurs whenever there are partially transmitting objects or the light pulse hits an edge. The returns are typically ordered by distance or by the characteristic of the target reflection.

Angle and Layer

To generate a 3D image of the environment, each individual distance measurement is spatially deflected while the elevation angle θ and azimuth angle φ are determined. The sensor types used in this work obtain the horizontal spacial deflection by either rotating sender and receiver components (Velodyne) or a deflection mirror (Ibeo) [20–24]. With Velodyne sensors, the vertical Field of View (FoV) is obtained by stacking multiple receiver and transmitter components in order to obtain one transmitter and receiver couple per layer. The Ibeo sensor, employ one transmitting unit for multiple layers and multiple receiving elements for height differentiation. The elevation angle θ is typically also referred to as the layer and is determined on the basis of the vertical alignment of the sender and receiver components. The azimuth angle φ is determined by the known rotation of the sensor components themselves or the known deflection mechanism, which is why the spherical coordinates $\mathbf{p} = (p_d, p_\theta, p_\varphi)$ are measured directly by the sensor. For further processing steps, those coordinates are often transformed into Cartesian coordinates by

$$p_x = d \cdot \sin \theta \cdot \cos \varphi \quad (2.2)$$

$$p_y = d \cdot \sin \theta \cdot \sin \varphi \quad (2.3)$$

$$p_z = d \cdot \cos \theta \quad (2.4)$$

which results in the Cartesian point vector $\mathbf{p}_{cart} = (p_x, p_y, p_z)$.

Intensity and Echo Pulse Width

In addition to the 3D information, LiDAR sensors typically provide information about the backscattered light signal. In general, and depending on the sensor components used, it is technically possible to sample the entire backscattered signal, which yields a large amount of information about the scanned environment. State-of-the-art LiDAR sensors typically provide an estimation of the length or height of the reflection signal, as shown in Figure 2.3. The intensity is defined as the height of the pulse above the threshold P_{th} , typically corrected over the distance in order to provide a distance independent measure for the backscattered intensity [25]. The EPW is defined as the length of the pulse above the threshold P_{th} . Multi-return LiDAR sensors often use the intensity or

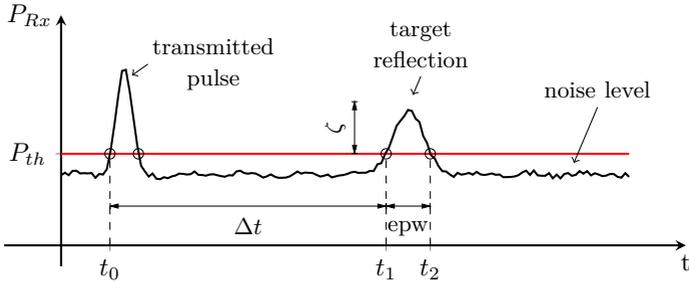


Figure 2.3: Schematic representation about the transmitted and backscattered light signal from a target reflection. The intensity ζ and EPW are the height and length of the received pulse above the threshold P_{th} . The time of flight is denoted as Δt .

EPW to further distinguish among multiple returns in addition to the distance information, for example by providing the strongest return in terms of intensity and first return in terms of measured distance [25].

2.1.2 Data Representation

Point Cloud

The raw data from LiDAR sensors is commonly presented in the form of a *point cloud*. A point cloud \mathbf{P} is defined as a list of point vectors \mathbf{p} . Each point vector contains at least Cartesian or spherical coordinates and can be expanded to include additional attributes. Subsequent data processing - e.g. algorithms for environment perception - are usually based on the point cloud data format.

The properties and structure of the point cloud reflect the sensor principle, which requires algorithms to be adapted to different sensor principles. Typical structures of point clouds obtained by LiDAR sensors with rotating transmitter and receiver components include, for example, the characteristic circles per layer, e.g. of ground-based reflections. By way of example, Figure 2.4 shows a point cloud obtained by a rotating sensor concept with 32 vertically stacked sender and receiver modules. The vertical resolution of such sensor concepts is also quite limited and typically less than a few hundred pixels (i.e. layers). The resulting point cloud is therefore referred to as a *sparse* point cloud. These

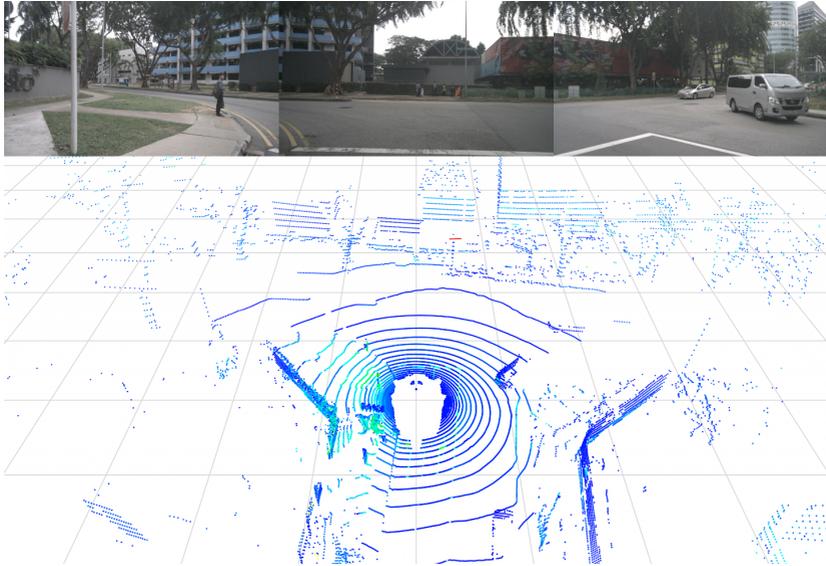


Figure 2.4: Example 3D point-cloud of a Velodyne HDL-32E [21]. For each point, the received intensity is displayed as a color code (dark=low intensity, bright=high intensity). At the top of the image, there are three corresponding camera images shown. The point-cloud data and images are taken from scene 125 of the NuScenes dataset [26].

characteristics are not evident in point clouds obtained by sensors with a flash exposure of the entire **FoV** (e.g. by depth cameras), where the resulting point cloud provides a *dense* representation of the environment.

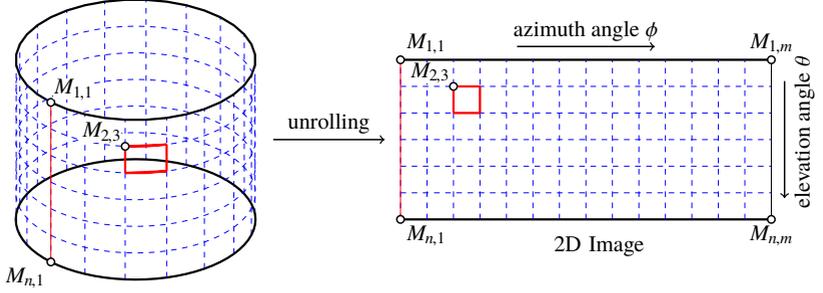


Figure 2.5: Principle of unrolling 3D LiDAR scans to a 2D representation of the image. The diagram is based on [27].

Point List

The raw data from state-of-the-art LiDAR sensors is defined as a point cloud $\mathbf{P} \in \mathbb{R}^2$, where each row $i = (1, \dots, n)$ contains one point with $j = (1, \dots, m)$ number of attributes.

$$\mathbf{P}^{n \times m} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix} \quad (2.5)$$

LiDAR sensors commonly provide 3D data of the environment in spherical coordinates with the radius d , azimuth angle φ and elevation angle θ , often combined with additional information about the characteristic of the backscatter signal, typically an estimated intensity ζ or the width of the returned pulse, which is referred to as EPW. This information is represented in each column of the point cloud. In the literature, point clouds are also referred as point lists and the elevation angle θ is also referred to as layer. In this work, the term "point lists" denotes an unordered list of points.

Lidar 2D Images

LiDAR 2-Dimensional (2D) images are matrices $\mathbf{M} = (m_{i,j}) \in \mathbb{R}^{(n \times m)}$, where each row i represents one elevation angle θ and each column j one azimuth angle ϕ . The point cloud is therefore structured as an image in which each pixel is located in the corresponding position within the field of view of the sensor. [Figure 2.5](#) illustrates the unrolling of a 360° **LiDAR** frame into a **2D** image representation. For each attribute of the point cloud, there is one corresponding **LiDAR** image matrix - for example, the distance matrix $D \in \mathbb{R}^{(n \times m)}$ or intensity matrix $I \in \mathbb{R}^{(n \times m)}$. This representation corresponds to the usual representation of images, whereas instead of stacked color channels (e.g. with RGB values), the **LiDAR** image contains stacked channels with distance, intensity or further attributes.

2.2 Scattering and Absorption

For a better understanding of weather effects on **LiDAR** sensors, we first need to examine how an electromagnetic wave interacts when it encounters particles in the air. This section therefore describes scattering and absorption, covering propagation through atmospheric particles. It is based on the research of [28, p. 49f ff] and [29].

2.2.1 Propagation in the Atmosphere

Since rain and fog consists of water particles, the fundamental optical properties are identical and so can be treated analogously, taking into account their respective droplet size and distribution. Their shape is spherical or can be parameterized as spherical with the radius r . When the electromagnetic wave that propagates through a homogeneous medium such as air encounters a particle, some of the energy is absorbed by the particle and dissipated into heat, and the remainder is scattered. For the wavelength λ , at a distance

$$R > \frac{r^2}{\lambda} \quad (2.6)$$

the wave front can be described as spherical. The power flow subtracted from a plane wave propagating through a layer of randomly distributed particles is calculated by the integral of the contributions of the individual scatters, as shown by [30]

$$\beta(\lambda) = 10^{-3} \int_{r_1}^{r_2} C_{ext}(r, \lambda) n(r) d(r) \quad (\text{neper/km}). \quad (2.7)$$

The **particle size distribution (PSD)** is defined as $n(r)$, representing the number of particles per unit volume and radius increment. The C_{ext} defines the extinction cross-section of the particle. The volume extinction coefficient $\beta(\lambda)$ is measured in neper/km, but is often expressed in dB/km, where $1 \text{ dB/km} = 4.343 \text{ neper/km}$.

The attenuation (i.e. decrease in energy transmission) experienced by a wave along the wave propagation in the z -direction through a PSD $n(r)$ is described by the Bouguer-Lambert-Beer law:

$$I(z) = I_0 \exp(-\beta z). \quad (2.8)$$

The incoming energy flux $I(z)$ is specified as $z = 0$ with I_0 and the specific attenuation with β . Equation 2.7 and Equation 2.8 are based on two assumptions: (1) *Scattering is independent*, where the interaction between the incident wave and each particle is regarded without taking into account the scattering of other particles and; (2) *multiple scattering effects are negligible*.

With the introduction of the concept of transitivity, the fraction of energy transmitted $T(z)$ is defined as [28]

$$T(z) = \frac{I(z)}{I_0} = \exp(-\beta z). \quad (2.9)$$

2.2.2 Electromagnetic Scattering

The Mie theory defines an analytic expression of the electromagnetic field scattered by a sphere within a homogeneous and isotropic medium when the particle is illuminated by a uniform plane wave [30]. Defining the dimensionless extinction efficiency $Q_{ext} = C_{ext}/\pi r^2$, the Mie theory takes the form of a series

$$Q_{ext} = \frac{2}{k^2} \sum_{m=1}^{\infty} (2m+1) \operatorname{Re}\{a_m + b_m\} \quad (2.10)$$

where $k = 2\pi r/\lambda$ is the dimensionless diffraction parameter and Re is the real part of the sum of the expansion coefficients a_m and b_m . The extinction efficiency Q_{ext} is therefore defined as the sum of scattering Q_{sca} and absorption Q_{abs} efficiency [31]:

$$Q_{ext} = Q_{sca} + Q_{abs} \quad (2.11)$$

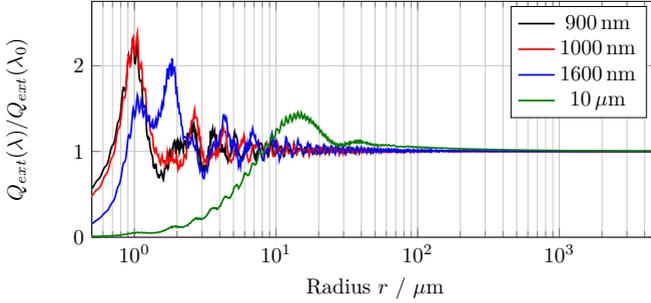


Figure 2.6: Ratio between the extinction efficiencies $Q_{ext}(\lambda)/Q_{ext}(\lambda_0)$ with $\lambda_0 = 550$ nm as a function of droplet radius r at different optical wavelengths λ . The image is based on [28]; the complex refractive index is taken from [32]; and the calculation is performed by the implementation of [33].

Extinction Efficiency

The sensitivity of the Q_{ext} is evaluated for typical LiDAR wavelengths in the near-infrared range and for the particle size of water droplets in fog and rain. The complex refractive indexes $n = v - ik$ are a function of the frequency of the incoming wave and describe the electromagnetic properties of particles with a given size and shape. Through calculation of the ratio of $Q_{ext}(\lambda)/Q_{ext}(\lambda_0)$ as a function of droplet radius, the sensitivity of Q_{ext} is comparable to human perception with λ_0 at 550 nm. The value for λ_0 is equal to the peak of the optical transfer function of the human eye at 550 nm [28]. This comparison is represented in visual form in Figure 2.6 for the typical wavelengths used for LiDAR sensors and the water droplet sizes that typically occur in rain and fog. The complex refractive indexes of water are taken from [32].

Analysis of the ratio $Q_{ext}(\lambda)/Q_{ext}(\lambda_0)$, shows that the extinction efficiencies of the evaluated wavelengths are oscillating for fog particles with a typical radius of 1 – 10 μm [28, 29]. For 900 and 1000 nm, the extinction efficiency reaches its maximum at approximately two, which means that the extinction efficiency is doubled compared with the baseline wavelength λ_0 . For larger wavelengths ($\lambda > 1500$ nm), the oscillation amplitude decreases progressively. At a wavelength of 10 μm the quotient is close to zero, which means that the extinction efficiency is significantly smaller than the base wavelength λ_0 efficiency.

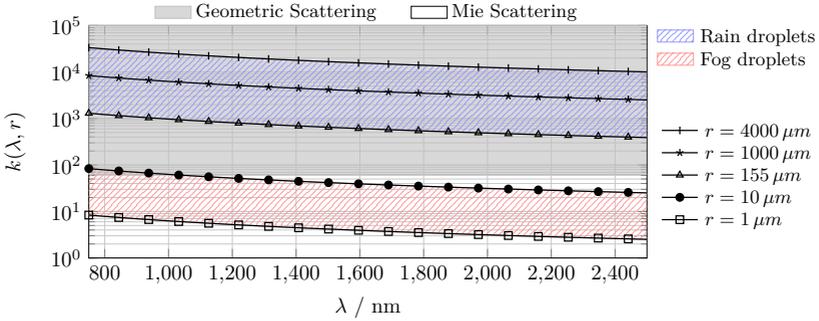


Figure 2.7: Diagram showing asymptotic convergence for electromagnetic back-scattering. The case distinction is shown for the ratio of droplet size radius r and wavelength λ in the near-infrared range.

For water droplets in rain with a typical radius of $155 \mu\text{m} - 4 \text{ mm}$, the extinction coefficient ratios are close to 1, which means that the extinction efficiencies are approximately the same for the given wavelength λ and λ_0 [28, 29].

Asymptotic Convergence

In mathematics, asymptotic convergence characterizes the behavior of functions when the independent variable approaches a certain value [34]. This concept is applied to the scattering (Equation 2.10) for the diffraction parameter k , which defines the ratio between the circumference of a particle $2\pi r$ and the wavelength λ . If the wavelength λ is much smaller than the particle (e.g. $r < 0.05\lambda$), the Rayleigh theory holds and $Q_{ext} \propto \frac{r^4}{\lambda^4}$. For large droplets compared with the wavelength (e.g. $r > 10\lambda$), geometric scattering with the classical laws of diffraction, reflection and refraction applies. Otherwise, the Mie theory with Equation 2.10 remains without simplification.

The application of this case distinction in relation to the size of water droplets in fog and rain is shown for the near-infrared range, which is where LiDAR sensors typically operate (see Section 2.1): Figure 2.7 relates the radius of water droplets from rain and fog to the wavelength of LiDAR sensors, indicating which scattering mechanism dominates. Water droplets in rain typically measure $155 \mu\text{m} - 4 \text{ mm}$, while those in fog typically measure $1 - 10 \mu\text{m}$ [28, 29].

Figure 2.7 therefore shows that geometric scattering can be applied for rain, while Mie scattering can be applied for fog with large droplets. This result is also reflected in Figure 2.6 and was therefore to be expected. While the influence of rain might be simpler to simulate than the influence of fog since only geometric scattering is present in rain, Mie and geometric scattering can also occur in fog, depending on the droplet size.

2.3 Weather

To study the influence of weather - especially of rain and fog - on LiDAR sensors, it is best to first provide a definition of these conditions. Quantification measures to distinguish between different intensity levels of rain and fog are therefore of particular interest. This section is based on [16, 28, 29, 35, 36], to which reference is made for further elaboration.

2.3.1 Fog

The following section provides a definition of fog and then analyzes the influence on the measuring principle of LiDAR sensors on the basis of the microphysical properties. The meteorological classification of different fog levels is then presented.

Definition

Fog is generally defined as suspended water droplets in the atmosphere that are close to the earth's surface and reduce visibility to below 1 km. [28, 37]. Fog typically appears with a relative humidity close to 100 %, whereas *mist* is an intermediate condition in which the relative humidity is above 60 % and visibility is greater than 1 km [28]. Any suspension of particles that are solid, dry and of microscopic size (e.g. smoke, sand, dust) is defined as *haze* [28]. Fog differs from cloud only in that it appears close to the earth's surface while clouds form at some distance above the surface [37].

The distribution of droplet sizes and the concentration of cloud condensation seeds determines the extent to which visibility is reduced [37]. Saturated water vapor and, in turn, fog occurs either when the air cools or water vapor increases. The first type can be divided into the following groups [28, 38] and [28, 38, 39, p. 115 ff.]:

- *Radiation fog*: Radiation cooling of the earth's surface lowers the air temperature to such an extent that relative humidity is saturated. In continental areas, this characteristic is usually observed at night during colder seasons with moderate wind conditions. Radiation fog has smaller water droplets and a lower liquid water content than advection fog.
- *Advection fog* arises when a warm and humid air mass moves over a colder surface. It can be further categorized into marine fog, which is caused by the advection of sea air from warm to cold oceanic areas, and coastal or sea fog, which forms when warm air moves from the sea to the inland.
- *Upslope fog* is caused by a decrease in the temperature of an air mass, which is moved upward by the wind along the slopes of hills or mountains.

Steam and frontal fog are caused by an increase in atmospheric water vapor. In general, polluted air results in a higher number of small particles than clean air.

Microphysical Characteristic

The microstructure characteristic of fog (i.e. the PSD) is summarized on the basis of the research of [28, 29, 36, 40]. Compared with cloud, fog has a relatively low water content, small droplets and small droplet concentrations per unit volume. The droplets can be assumed to be spherical in shape with a typical diameter of $1 - 10 \mu\text{m}$ [28] or between $2.5 \mu\text{m}$ and a few tens of micrometers [29].

The distribution of the volume concentration of droplets per unit radius as a function of the radius r is approximated by a modified gamma function, as first proposed by Deirmendjian [41]:

$$n(r) = Ar^\alpha \exp\left[\frac{\alpha}{\gamma}\left(\frac{r}{r_c}\right)^\gamma\right] \quad (\text{cm}^{-3}\mu\text{m}^{-1}). \quad (2.12)$$

The constants A , α , γ and r_c define the shape of the distribution of $n(r)$. The total number of water droplets N per unit volume is defined by

$$N = \frac{A}{\gamma b^{(\alpha+1)\gamma}} \Gamma\left(\frac{\alpha+1}{\gamma}\right) \quad (\text{cm}^{-3}) \quad (2.13)$$

and the total liquid water content by

$$W = \frac{4}{3}\pi 10^{-6} p_w \frac{A}{\gamma b^{(\alpha+4)\gamma}} \Gamma\left(\frac{\alpha+4}{\gamma}\right) \quad (\text{g m}^{-3}). \quad (2.14)$$

The liquid water density is given by p_w (g cm^{-3}) and $b = \frac{\alpha}{\gamma r_c^\gamma}$ in ($\mu\text{m}^{-\gamma}$) [28]. With $\gamma = 1$, equation 2.12 corresponds to a gamma distribution, which could also be expressed by an logarithmic normal distribution [29, p. 26].

Specific Attenuation

Equations 2.7, 2.10 and 2.12, require extensive information about the physical properties of fog in order for the attenuation β to be calculated. In [28, pp. 57 ff.], this calculation is performed for different types of fog, using various PSD parameters available in the relevant literature. However, the quantification of fog parameters according to the microphysical model is impractical due to the lack of availability of particle distributions and the temporal and spatial variety of the properties. Empirical models are therefore required for predicting attenuation in fog [42–44].

Kruse originally proposed an empirical model that relates the visibility range in km with the fog attenuation by

$$V(\lambda) = \frac{10 \log_{10}(T)}{\beta} \left(\frac{\lambda}{\lambda_0}\right)^{-q} \quad (2.15)$$

where λ_0 is the maximum spectrum ($\lambda_0 = 550$ nm) of the solar band and T is the visual threshold and defined as 2% [42, 43]. The coefficient q is related to the distribution of the fog particles and was originally defined by [43] and optimized by [44]:

$$q = \begin{cases} 0 & \text{if } V < 0.5 \text{ km dense fog} & (2.16) \\ V - 0.5 & \text{if } 0.5V < 1.0 \text{ km moderate fog} & (2.17) \\ 0.16V + 0.34 & \text{if } 1.0 < V < 6.0 \text{ km mist} & (2.18) \end{cases}$$

However, the assumption that β is independent of the wavelength for fog with $V < 0.5$ km is disproved by experimental data for selective wavelengths at 830 nm and 1.55 μm by [45, 46]. The research of Ijaz et al. further defines the wavelength dependency of q by evaluating experimental data captured in fog and smoke. The coefficient q is therefore described as a linear function of wavelength

$$q(\lambda) = 0.1428\lambda - 0.0947 \quad (2.19)$$

which is valid for the near-infrared range ($0.55 < \lambda_0 < 1.6 \mu\text{m}$) and visibility range 0.015–1.0 km [42]. Accordingly, the attenuation in fog can be calculated by

$$\beta(\lambda) = \frac{10 \log_{10}(T)}{V} \left(\frac{\lambda}{\lambda_0} \right)^{-q(\lambda)} \quad (\text{dB/km}). \quad (2.20)$$

Meteorological Visibility

The basis for meteorological visibility was defined in 1924 by Koschmieder, who proposed a theory concerning the apparent luminance of a black object C_x that can be distinguished from the background C_0 [16, 35, 47]

$$C_x = C_0 e^{-\beta z}. \quad (2.21)$$

Meteorological Optical Range

Since the meteorological visibility is based on subjective factors, a more objective measure based on the transparency of the atmosphere is represented by the **meteorological optical range (MOR)** [47]. The **MOR** is defined as the length

of the path in the atmosphere over which the luminous flux in a collimated light beam with a wavelength of 550 nm from an incandescent light source is reduced to 5%. The color temperature is defined here as 2.700 K [47]. The wavelength of 550 nm equates to the peak of optical transfer function of the human eye [28]. In accordance with the Bouguer-Lambert-Beer law (2.8), the MOR can be written as:

$$0.05 = \frac{I}{I_0} = e^{-\beta z} \quad (2.22)$$

and results in

$$z = -\frac{\ln 0.05}{\beta} \approx \frac{3}{\beta}. \quad (2.23)$$

The link to the attenuation expressed in dB/km by the empirical fog model (equation 2.15) is given by

$$\beta_{10} = -\frac{10 \log_{10}(0.05)}{z} \quad (\text{dB/km}) \quad (2.24)$$

with $q = 0$ and the atmospheric attenuation β_{10} in decibel per unit length $r = 1$ [48]

$$\beta_{10} = -10 \log_{10}(T(1)) = 10 \log_{10}(e)\beta. \quad (2.25)$$

The application of equation 2.25 in 2.24 and the logarithmic laws results in the definition of the MOR in 2.23:

$$z = -\frac{10 \log_{10}(0.05)}{10 \log_{10}(e)\beta} = -\frac{\ln(0.05)}{\beta}. \quad (2.26)$$

The MOR results in the meteorological visibility, when the magnitude of the apparent contrast of a black object seen against the horizon C_x/C_0 is 0.05. Equation 2.21 then reduces to $0.05 = e^{-\beta z}$, which is the MOR at the distance z .

A distinction among the different weather conditions based on the MOR is defined in the International Visibility Code (IVC) shown in Table 2.1.

Table 2.1: International visibility definition based on [49].

Code No.	Weather Condition	MOR
0	Dense Fog	< 50 m
1	Thick Fog	> 50 m and < 200 m
2	Moderate Fog	> 200 m and < 500 m
3	Light Fog	> 500 m and < 1 km
4	Thin Fog	> 1 km and < 2 km
5	Haze	> 2 km and < 4 km
6	Light Haze	> 4 km and < 10 km
7	Clear	> 10 km and < 20 km
8	Very Clear	> 20 km and < 40 km
9	Exceptional clear	> 50 km

2.3.2 Rain

This subsection defines rain as a weather condition and analyzes its impact on the measuring principle of **LiDAR** sensors on the basis of microphysical properties. This subsection also presents the meteorological classification of different rain intensities.

Definition

Rain is defined as precipitation in the form of liquid water droplets with a diameter greater than 0.5 mm, while drizzle is defined as precipitation in the form of liquid water droplets with a diameter less than 0.5 mm [50]. Drizzle generally contains many more water droplets and attenuates visibility more than light rain does.

Microstructure Characteristic

Rain consists of deformed water droplets, which can be parameterized by the equivalent radius r_0 defined as the radius of a sphere with the same volume as the deformed droplet [29]. The deformation of rain droplets depends on the radius r . For a radius $r \lesssim 140\mu\text{m}$ droplets are nearly perfect spheres when falling at terminal velocity. Larger drops with $140 \lesssim r \lesssim 500\mu\text{m}$ are slightly deformed; at a radius greater than $r \gtrsim 500\mu\text{m}$ the droplets take the

form of flattened spheroids. At a radius greater than $r \gtrsim 5\text{mm}$ droplets are hydrodynamically unstable and decompose [29].

An analytical approximation of the terminal velocity of falling drops is a function of droplet radius and reaches its maximum at about 9 m/s for a radius of 2 – 3 mm [28]

$$v(r) = 9.65 - 10.3 \exp(-1.2r) \quad (\text{m/s}). \quad (2.27)$$

The PSD of rain is analytically described as a gamma distribution

$$n(r) = N_0 r^\mu e^{-\gamma r} \quad (\text{m}^{-3} \text{mm}^{-1}) \quad (2.28)$$

where μ is the shape parameter ($-3 \lesssim \mu \gtrsim 8$). N_0 and γ are expressed by

$$N_0 = 6 \cdot 10^4 e^{(3.2 - \ln 5)\mu - \ln 5} \quad (\text{m}^{-3} \text{mm}^{-1-\mu})$$

$$\gamma = 0.2 \left[\frac{R}{33.31 N_0 5^{1+\mu} \Gamma(4.67 + \mu)} \right]^{-\frac{1}{4.67+\mu}} \quad (\text{mm}^{-1}). \quad (2.29)$$

Furthermore, based on the terminal velocity and PSD, the rainfall rate R results in

$$R = 4.8\pi \cdot 10^{-3} \int_{r_{min}}^{r_{max}} r^3 v(r) n(r) dr \quad (\text{mm/h}). \quad (2.30)$$

Specific Attenuation

In the near-infrared range, attenuation due to rain is almost completely independent of the wavelength [28, p. 63]. The wavelengths are much smaller than the rain droplets, which is why asymptotic formulas for large particles can be applied as presented in Section 2.2.2.

Rainfall Rate

Precipitation intensity describes the amount of precipitation - or, more precisely, the amount of precipitation that falls over an area of m^2 in mm over a

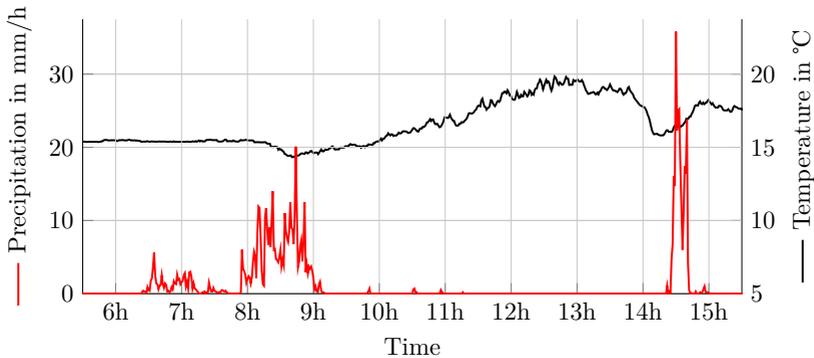


Figure 2.8: Precipitation intensity and air temperature captured on a summer’s day in southern Germany in August 2019.

given time unit [51]. Based on the rainfall rate R (usually given in mm/h), the precipitation can be classified as rain or drizzle. A definition is given in [50] and [51] (see Table 2.2). As an example of naturally occurring precipitation intensities, Figure 2.8 shows the rainfall on a summer’s day in southern Germany in 2019. Between 14h and 15h the intensity reaches a maximum level of 35.8 mm/h and remains above 1.7 mm for more than 10 minutes, which means that it can be classified as *strong* rain based on the definition in Table 2.2 [51].

2.4 Weather Simulation in Controlled Environments

As Table 2.1 and Table 2.2 show, there is a significant variance in weather conditions. The precipitation curve in Figure 2.8 shows how dynamically the rain intensity can shift and, in particular, how quickly very high intensities can be observed. It is therefore extremely difficult to detect such weather conditions dynamically, especially in on-road scenarios, which is why controlled environments are required for reproducibly simulating such conditions.

The state-of-the-art climate chambers used in the automotive industry are designed primarily for the purpose of conventional vehicle-centric testing of

Table 2.2: Precipitation levels for rain and drizzle based on [50, 51].

Precipitation Type	Precipitation Level in mm/60min	Precipitation Level in mm/n min
Rain [50]		
light	< 2.5 mm (0.1 in.)	and < 0.25 mm (0.01 in.) with $n = 6$ min.
moderate	≥ 2.6 mm (0.11 in.) and < 7.6 mm (0.3 in.)	and < 0.76 mm (0.03 in.) with $n = 6$ min.
heavy	> 7.6 mm (0.3 in.)	or ≥ 0.76 mm (0.03 in.) with $n = 6$ min
Rain [51]		
light	< 2.5 mm	and < 0.5 mm with $n = 10$ min
moderate	≥ 2.5 mm and < 10.0 mm	and ≥ 0.5 mm and < 1.7 mm with $n = 10$ min
strong	≥ 10.0 mm	and ≥ 1.7 with $n = 10$ min
very strong	≥ 50.0 mm	and ≥ 8.3 with $n = 10$ min
Drizzle [51]		
light	< 0.1 mm	
moderate	≥ 0.1 mm until < 0.5 mm	
strong	≥ 0.5 mm	

the influence of weather and temperature on vehicles and their components. The size and shape of these systems is therefore designed to ensure that vehicles are positioned centrally and without a large amount of free space around them. This can be seen in [Figure 2.9](#), which shows two examples of the kind of climate chamber used in the automotive industry [52, 53]. These examples show that insufficient space is available - especially in front of and around the vehicle - to investigate the performance of sensors potentially used in automated or autonomous systems. Due to their limited size and shape, therefore, such climate chambers are not suitable for analyzing sensor perception systems designed for autonomous driving systems.

Moreover, conventional climate-wind chambers are not designed for reproducing natural rainfall or natural fog, but rather for exposing a vehicle to a defined amount of water combined with high wind speeds corresponding to the headwind of the velocity to be simulated. These tests are performed over the entire temperature profile of a conventional passenger car (e.g. -40° to $+60^{\circ}$ [52] or -20° to $+55^{\circ}$ [53]). Weather conditions such as homogeneous fog are not considered to be of major importance and so cannot be generated reproducibly, which means that the conventional climate chamber used in the automotive industry are not suitable for the in-depth analysis of sensor perception in adverse weather conditions. This work is therefore based on the results from dedicated



Figure 2.9: Examples of automotive manufacturer climate chambers [52,53]

climate chambers specially designed to study human and machine perception in adverse weather conditions.

The information relating to different fog and rain simulators in the two following subsections is based on the work of [54] and supplemented by information from [55].

2.4.1 Artificial Rainfall

The production of artificial rain is the subject of intense research in the agricultural sciences [56–59]. Artificial rain is usually produced by means of nozzles mounted on a metal structure a few meters above the ground; the number of nozzles, area covered and height are variable. The height typically varies between 1 m and 7 m and determines the velocity of the generated raindrops. The number of nozzles and the area covered depends on the application and is typically below a few tens of square meters. Furthermore, the rain droplet distribution, rainfall rate and rain drop velocity can be influenced by the choice of nozzle and the water injection pressure [54]. In [60], the droplet distribution is altered though the attachment of a mesh under the nozzles. For the purpose of analyzing sensor perception capability for autonomous driving, the area covered by rain in these systems is usually too small, which means that the result cannot be applied. For the analysis of suitable rain simulators, we use the work of [54], which lists and evaluates suitable simulations with large rain areas, supplemented by the Carissima Chamber in Ingolstadt [55].

Since no publicly available technical details about the rain simulator used by the **Public Works Research Institute (PWRI)** is publicly available, we use the information given in the work of [61], which was recorded in the **PWRI** simulator. Hence, the **PWRI** is capable of producing rainfall rates of 20 – 100 mm/h and fog visibility of 10 – 200 m [61].

The **Virginia Smart Roads (VSR)** facility is capable of producing rain with 2 – 63 mm/h, although no details of the droplet size or distribution are available [54]. In the chamber of [62], the simulated rainfall rate is between 10 – 150 mm/h without a homogeneous distribution of the water amount [54]. In addition, the Leibniz University Hanover operates a rain simulator with a uniform precipitation between 9.2 and 98.1 mm/h. However, the rain area of 4 m by 3 m is too small for dynamic automotive experiments [54,63]. A large rain area of 5 m by 50 m is available in the Carisma chamber in Ingolstadt, Germany [55]. The rainfall rate is expected to be between 14 and 88 mm/h.

2.4.2 Artificial Fog

To analyze the influence of fog on **LiDAR** sensor perception under controlled environmental conditions, the simulation requires a highly realistic droplet size distribution, spatial uniformity and temporal stability, because the backscattered light depends on the distribution and size of the water droplets [54]. According to Colomb et al., who have spent many years researching the simulation of natural fog, only three facilities capable of producing fog for automotive applications are available: the **VSR** in the United States, the **PWRI** in Japan and Cerema in France [54,62,64]. Since the **VSR** is an outdoor facility, the requirement of spatial uniformity is not fulfilled [65], although the length of 800 m offers a range of opportunities for test scenarios and realistic setups without any limitation in maximum viewing range, especially for camera or **LiDAR** sensors.

The length of the indoor facility in Japan (**PWRI**) is 200 m, considerably more than that in France (31 m). To our knowledge, however, the Cerema facility in France is the only one capable of controlling droplet size distribution, as well as the spatial and temporal stability of the simulated fog [62]. Note that at the time this work was created, no fog simulation facilities were in operation in

Table 2.3: Climate chambers for producing artificial rain or fog, mostly with the purpose of sensor testing for autonomous driving [54].

Name	Location	Size (b,l,h)	Rain Simulation	Fog Simulation	Additional Information
PWRI	Japan	9.8x200x6.9 m ³	20 – 100 mm/h	10 – 100 m	
VSR	USA	10x800x10 m ³	2 – 63 mm/h	3 – 90 m	
Cerema	France	5.5x31x2.4 m ³	10 – 150 mm/h	5 – 200 m	spatial (only for fog) and temporal uniformity, droplet size distribution is measured
LUH	Germany	4x3x3 m ³	9.2 – 98.1 mm/h	–	spatial uniformity, droplet size distribution is measured
Carissima	Germany	120x30 m ³	14 – 88 mm/h	–	

the Carissima climate chamber in Ingolstadt [55]. This facility was therefore not considered.

Table 2.3 contains a summary of all the artificial fog and rain simulators and an overview of the meteorological visibility and rainfall that can be simulated.

2.5 Machine Learning in Computer Vision

This section covers the basic principles of machine learning algorithms based on [66–71], with an emphasis on the approaches applied in this work. For a complete overview of machine learning methods and for further in-depth study, the information provided in [66–68, 70] is recommended.

Machine learning can be defined as the design of algorithms that automatically learn through experience [69]. The term "experience" here usually refers to the process of *training data* on the basis of which machine learning algorithms can build a mathematical model for making decisions or predictions without being implemented explicitly [70, 72]. Machine learning approaches can be categorized as *unsupervised* and *supervised* learning.

Unsupervised learning methods are not based on labeled training data, but instead learn the structure of given input data. The goal is to gain new insights based on the learned structures within the input data [67, p.9]. The absence of labeled training data is a major advantage, particularly when very large amounts of data are processed.

Supervised learning methods learn from the basis of labeled training data by optimizing the function to predict the corresponding ground truth labels from the input data. The aim of the function is to predict the correct class label of an unseen input data sample, which is defined as generalization [67, p.276].

Furthermore, semi-supervised learning methods are based on combinations of both, whereby, for example, only parts of the database are manually annotated, annotations are generated without manual intervention or the algorithm can itself request a user label for a new data point (active learning).

Reinforcement learning involves evaluating system actions or a series of system actions, whereby correct behavior leads to positive rewards and incorrect behavior leads to negative rewards [67, p.9]. Using reinforcement learning, for example, an algorithm can learn a board game solely on the basis of the rules of the game, the conditions for victory and by playing intensively against itself [71]. Since the reward influences the learning progress by providing feedback, this method is also considered to be a supervised method.

For the purposes of this work, we will focusing on supervised methods, which comprise two key task types: (1) *classification* and (2) *regression*. *Classification*, as the name suggests, typically refers to the categorization of data into either concepts or collections that apply the concept of family similarities. In the context of images and machine learning, classification at pixel level is often referred to as segmentation. *Regression* typically involves the analysis of data trends in order to make predictions on the basis of these trends [68, p.121].

2.5.1 Feature Extraction

Machine learning approaches are usually based on two steps, feature extraction and the subsequent processing of the desired result. Feature extraction in this context is either manual or automated. Some examples of manual and automated feature extraction are outlined in the following.

Manually Extracted Features

Traditional approaches are based on two steps: First, features are derived manually from the input data; and, secondly, classification is performed by applying machine learning approaches. The first step is typically performed by an expert and results in a feature representation of the input data. An edge detection algorithm (e.g. Sobel operator) is one example of a well-known feature representation in the field of computer vision. In general, any representation of the input data that supports discrimination for the desired output is suitable as a feature representation. In general, any representation of the input data that supports discrimination for the desired output is suitable as a feature representation.

Feature Learning

Manual feature extraction can generally also be performed automatically, for example with a **p**rin**c**iple **c**omponent **a**nalysis (PCA), which is based on the eigenvector and eigenvalues of the covariance matrix of the input data [66]. By using the first n eigenvectors of the corresponding eigenvalues sorted in ascending order, the data is projected to a lower n -dimensional representation

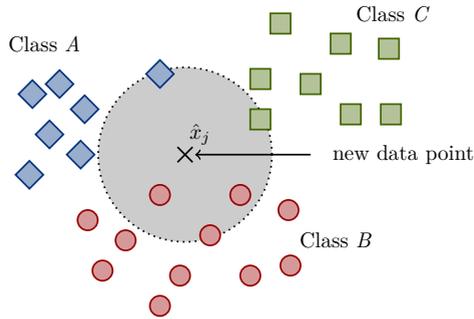


Figure 2.10: Diagram of a k NN algorithm for three classes with $k = 5$, i.e. regarding the $k = 5$ nearest neighbors.

without taking into account the target class label. Other unsupervised feature learning techniques are autoencoders, deep belief networks and **independent component analysis (ICA)**. One of the main drawbacks of unsupervised feature learning methods is that they do not take into account the class label in the learning process. Methods that include the class label and learn feature extraction under supervision have shown superior results, as demonstrated by the success of **CNN** in classification and object recognition [67, p.9].

2.5.2 Machine Learning Methods

The following subsections present the machine learning methods used in this work; for a complete overview of available methods, refer to [66].

2.5.3 k Nearest Neighbor

The classification method **k nearest neighbors (kNN)** is based on the similarity of neighboring points. Through the manual definition of a distance measure for similarity, each point is classified based on its k neighboring points, assuming a classification task for the set of data $\{(x_i, y_i)\}$ with x_i features, y_i corresponding class labels and i data samples [68]. The **kNN** algorithm classifies a new data point \hat{x}_j by computing the distance to all other points x_i , selecting the k nearest

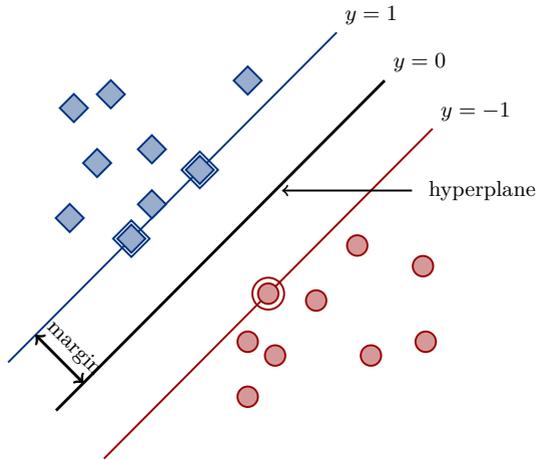


Figure 2.11: Diagram of an SVM for two classes. The support vectors are marked with double lines. This representation is inspired by [66, p. 327].

neighbors and applying their corresponding class labels y_i to the input data point \hat{x}_j . Figure 2.10 shows the classification for a new data point for its $k = 5$ nearest neighbors. The stored dataset $\{(x_i, y_i)\}$ can be regarded as training data, on which basis new data points are classified. The training procedure is performed simply by storing the input data, whereby the parameters k and distance function are set manually to fit to the existing data distribution. A generic example of a possible distance function is the Minkowski Distance

$$d(x, y) = \sqrt[p]{\sum_i (x_i - y_i)^p} \quad (2.31)$$

which becomes the Euclidean Distance for $p = 2$ [68].

2.5.4 Support Vector Machine

The term "maximum margin classifier" essentially describes the idea of an SVM classifier that determines a hyperplane between two linearly separable classes such that the distances between the two classes and the hyperplane

is maximal [66, p. 326 ff.]. This results in the mathematical optimization of maximizing the distance between the support vectors -, i.e. the points closest to the decision boundary - and the hyperplane itself. Figure 2.11 shows the decision boundary (hyperplane) between two classes with a maximized margin.

The basic mathematical representation of an SVM is based on the book by [66, p. 326 ff.]; for a complete overview of the mathematical optimization and for further details, refer to this chapter.

With the definition of the training dataset as input vectors \mathbf{x} as x_1, \dots, x_N with target values \mathbf{t} as t_1, \dots, t_N where $t \in \{-1, 1\}$ representing the classification label, each new data point x is classified based on the sign of $y(x)$ with

$$y(x) = w \cdot x + b. \quad (2.32)$$

Then the shortest distance between the hyperplane $y(x) = 0$ and each data point x_i is given by the perpendicular distance

$$\frac{|y(x)|}{\|w\|}. \quad (2.33)$$

Given that only the correctly classified points where $t_n y(x_n) > 0$ are of relevance, we obtain $t_n y(x_n) / \|w\|$. The maximum margin of the smallest distance between each data sample and the hyperplane will then result in the optimization:

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|} \min_n \left[t_n (w x_n + b) \right] \right\}. \quad (2.34)$$

Since $1/\|w\|$ is independent from n , the term could be written outside the optimization over n .

The application of the SVM to data distributions with non-linear class boundaries could be performed by introducing a fixed feature-space transformation $\phi(x)$ - the "Kernel" - that transforms the data to a higher dimensional space, where the classes are linearly separable [66, p. 326 ff.]. Since the input feature vector \mathbf{x} and kernel transformation are usually derived manually on the basis of the input data, the SVM is based on manually extracted features.

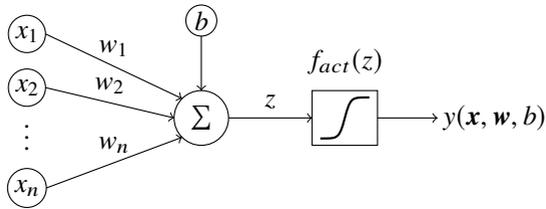


Figure 2.12: Diagram showing an artificial neuron with input vector \mathbf{x} , weight vector \mathbf{w} and bias b . The representation is inspired by [67, p. 63].

Furthermore, the [SVM](#) can be extended to deal with overlapping class distributions by allowing mis-classifications during optimization while also simultaneously penalizing them [66, p. 331 ff.].

2.5.5 Artificial Neural Networks

[CNNs](#) are a subcategory of [artificial neural networks \(ANNs\)](#). We will therefore discuss [ANNs](#) at first.

Elements of Neural Networks

An [ANN](#) consists of neurons and is powered by biological neural networks. Each neuron consists of dendrites, soma, nucleus, axon and synapses [67]. By describing the dendrites as input vector $\mathbf{x} \in \mathbb{R}^n$ and the synaptic strengths as weight vector $\mathbf{w} \in \mathbb{R}^n$, a nucleus can be mathematically described as

$$z = \mathbf{w}\mathbf{x}^T + b \quad (2.35)$$

where $b \in \mathbb{R}$ denotes the bias [67]. A nucleus therefore essentially calculates the weighted sum of its inputs and adds the bias. The result is then used as input for a nonlinear activation function $f_{act} : \mathbb{R} \rightarrow \mathbb{R}$. The resulting mathematical description of a neuron is

$$y(\mathbf{x}, \mathbf{w}, b) = f_{act}(\mathbf{w}\mathbf{x}^T + b). \quad (2.36)$$

[Figure 2.12](#) shows a computational graph diagram of a neuron.

Network Training

In neural networks, the term "training" is defined as the adjustment of weights \mathbf{w} and bias \mathbf{b} , which leads to an altered response of the neuron. By introducing $\tilde{\mathbf{w}}$, which contains the bias parameter b as an additional weight \tilde{w}_0 , the term "learning" refers simply to the adjustment of $\tilde{\mathbf{w}}$. Equation 2.35 and Equation 2.36 result in

$$\mathbf{y}(\tilde{\mathbf{x}}, \tilde{\mathbf{w}}) = f_{act}(\tilde{\mathbf{w}}\tilde{\mathbf{x}}^T) \quad (2.37)$$

where the input variable \tilde{x}_0 is set to $\tilde{x}_0 = 1$ [66, p. 225ff]. This definition simplifies the adjustment of weights and bias in one variable during the training process.

A mathematical description of network training aims to obtain a weight vector $\tilde{\mathbf{w}}$ that minimizes the error function $E(\tilde{\mathbf{w}})$. This minimization is typically achieved through gradient descent optimization. The sum-of-squares, for example, could be used as an error function. By describing the training process as a set of independent observations, the cross-entropy error function is given by the negative log likelihood

$$E(\tilde{\mathbf{w}}) = - \sum_{i=1}^n c_i \ln(y_i) + (1 - c_i) \ln(1 - y_i) \quad (2.38)$$

where y_i is one output neuron and n is the total number of output neurons (i.e. the number of classes) [66, p.235]. According to Simard et. al, the cross-entropy error function in multi-class classification tasks enables faster training and better generalization than the sum-of-squares [66, 73].

Activation Functions

An essential component of neural networks consists of the activation functions. A feed forward network containing only linear activation functions remains a linear function regardless of the number of neurons, which is why nonlinear activation functions are used to map non-linearities with neural networks. Furthermore, a differentiable activation function is required for gradient-descent-optimized learning. It is also desirable for an activation function to have a large gradient close to the origin, which means that convergence is accelerated

for input variables close to zero. The weight vectors and bias values are often initialized with near-zero values [67, p.71]. Figure 2.13 shows a selection of common activation functions and their derivatives. Activation functions like sigmoid or tangential hyperbolic fulfill all the described requirements, although the derivatives are saturating toward zero (see Figure 2.13a and Figure 2.13b) as the input value $|x|$ increases. During backpropagation in multi-layered neural networks, the multiplication of gradients close to zero brings learning progress to a standstill. These activation functions are therefore used only in networks with just a few layers [67].

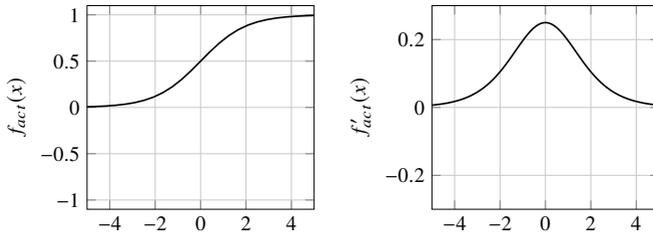
Rectified linear unit (**ReLU**) activation functions do not saturate as $|x|$ increases and are therefore recommended for neural networks with multiple layers. Furthermore, for non-biased input data, it is advantageous when non-linearity is close to zero. With the hyperparameter $\alpha \in \{0, 1\}$, they are defined as

$$f_{act}(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0. \end{cases} \quad (2.39)$$

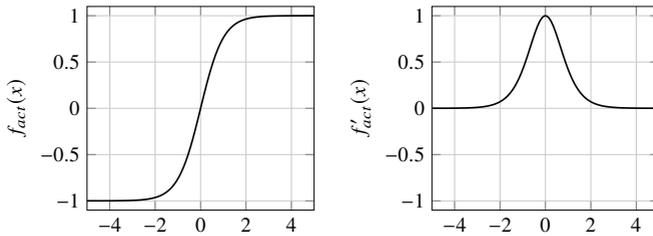
Mathematically, the **ReLU** function is not differentiable; the Exponential Linear Unit (**ELU**) function, however, can be seen as a smoothed variant of the **ReLU** function, which is differentiable [67]. When $\alpha = 0$, they are referred to as **ReLU**, for $\alpha > 0$ leaky **ReLU** [67]. Figure 2.13c and Figure 2.13d show a plot of **ReLU** functions with $\alpha = 0$ and $\alpha = 0.1$. The fast numerical computability of **ReLU** functions, is especially advantageous as the number of layers increases [67]. Accordingly, the **ReLU** activation functions have become established in recent years for deep neural networks and results in faster convergence of training as shown in [74, 75] and better result [76].

Network Topologies

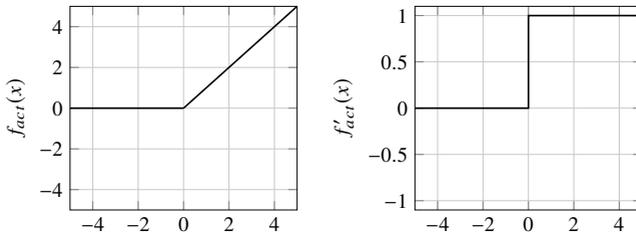
The linking of one or more neurons results in a neural network, which can exhibit a variety of typologies due to the large number of potential connections [67]. The first layer of an **ANN** without any prior layers is known as the "input layer". The last layer without any subsequent layers is known as the "output layer" and provides the result of the network, which can form any arbitrary nonlinear function by output neuron. All layers in between are known as "hidden layers". Figure 2.14 shows an **ANN** with three input neurons, two



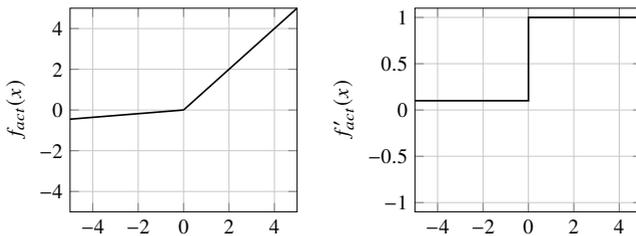
(a) Sigmoid function and its derivative.



(b) Tangent hyperbolic function and its derivative.



(c) ReLU function and its derivative.



(d) Leaky ReLU activation function and its derivative.

Figure 2.13: Selection of common activation functions and their derivatives.

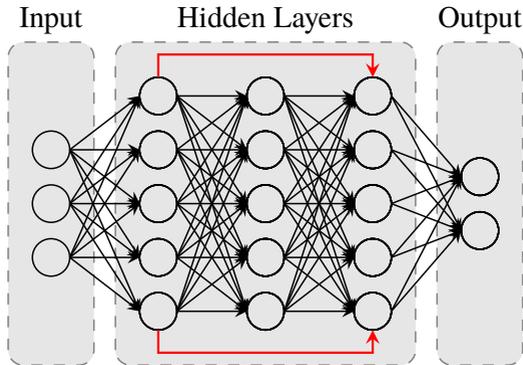


Figure 2.14: Diagram showing an FFN with three input neurons, two output neurons and three hidden layers. A skip or shortcut connection is shown in red.

output neurons and three hidden layers. Since each layer is linked only to the subsequent layer, the network topology is called **feed forward network (FFN)** [66, p. 225ff]. Figure 2.14 also shows two skip or shortcut connections, where one or more layers are skipped [77, p. 42]. Numerous other topologies like Boltzmann machines also exist, although these do not fall within the scope of this thesis; for further information on these, please refer to [68].

Output Layer

Assuming a binary classification problem with two classes C_1 and C_2 , the classification probability for each class is derived on the basis of the network output. For a network with one single output neuron and a logistic sigmoid activation function, the activation function of the output layer $y(\vec{x}, \vec{w})$ can be interpreted as the conditional probability $P(C_1|\vec{x})$. Consequently, $P(C_2|\vec{x})$ is given by $1 - P(C_1|\vec{x})$ [66, p. 235f.].

For a multi-class problem with n -classes and the binary coded variables $c_i \in \{0, 1\}$, which represent each class i with $c_i \in \{1, n\}$, n -output neurons are used according to the number of classes. Applying the softmax function

$$P_i(c_i|\tilde{\mathbf{x}}) = \frac{\exp(y_i(\tilde{\mathbf{x}}, \tilde{\mathbf{w}}))}{\sum_j \exp(y_j(\tilde{\mathbf{x}}, \tilde{\mathbf{w}}))} \quad (2.40)$$

the probability per class is derived from the result of each output neuron $y_i(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$ [66, p. 235f.]. Note that the activation function of the output layer is set to unit activation [66, 67].

2.5.6 Convolutional Neural Networks

The concept underlying CNN is based on the idea that, for example, an extracted feature is not affected by the position of an object in a given input space. In other words, it is irrelevant whether the object is on the right or left edge of the image.

Applying this idea to ANN indicates that a local environment is sufficient for training and the complete input space is not required in one training step. This local area in the input space is denoted as a receptive field and shifted multiple times to cover the complete input space, which means that the weights are shared over the input space. The number of weights can therefore be significantly reduced, is determined by the size of the receptive field and is independent of the input data size [66, p.268].

In addition to the sharing of weight over in the spacial domain of CNN architectures, recurrent neural network (RNN) architectures exist in which the output of a neuron is connected to one or more previous layers, which means that the weights are shared over the temporal domain. These architectures are often applied to time series data. In the field of autonomous driving, multi-modal sensor setups (e.g. camera, RaDAR and LiDAR) are used to perceive the environment, which apparently results in time series data. This time series data is then typically fused by bayesian filters (e.g. Kalman Filter), which assume independent measurements over time by each sensor component, in order to modal temporal correlations between measurements correctly. Perception algorithms based on RNN architectures have temporal dependencies and

are therefore not providing uncorrelated measurements over time. Thus, **RNN** architectures are not applied in this thesis and so will not be discussed in any greater depth. Detailed information on **RNN** is available in [66, 67, 77].

2.5.7 Evaluation Metrics

Metrics are essential for the evaluation of classifiers. Based on the *confusion matrix*, this section presents the metrics *accuracy* and *intersection over union (IoU)* are introduced in this section. For further metrics and details [67, p.106ff.], [78] and [68, p.121ff.] is recommended.

Confusion Matrix

The confusion matrix is defined as $\mathbf{M} = (m_{i,j}) \in \mathbb{R}^{(n \times n)}$ for n classes, where each element \mathbf{M}_{ij} represents the samples classified as class j that are actually in class i . Consequently, the elements on the diagonal \mathbf{M}_{ii} result in the correctly classified samples.

The notation of **true positive (TP)**, **true negative (TN)**, **false positive (FP)** and **false negative (FN)** are introduced with the confusion matrix of a binary classification problem:

$$\mathbf{M}^{2 \times 2} = \begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix}. \quad (2.41)$$

For multi-class evaluations, the observed class $c_k = \mathbf{M}_{11}$ is presented, whereby the predicted classification result is defined as c_p and the actual label (i.e. ground truth) is defined as c_l [79]:

- The **TP** samples are correctly classified with $c_p = c_l$ for the observed class $c_k = c_p$.
- The **FP** samples are falsely classified $c_p \neq c_l$ for the observed class $c_k = c_p$.
- The **FN** samples are falsely classified ($c_p \neq c_l$), but not the observed class ($c_k \neq c_p$).

		Predicted		
		$c_0 \dots c_{k-1}$	c_k	$c_{k+1} \dots c_n$
True Label	$c_0 \dots c_{k-1}$	TN	FP	TN
	c_k	FN	TP	FN
	$c_{k+1} \dots c_n$	TN	FP	TN

Figure 2.15: Confusion matrix for multi-class classification with n -classes. The terminology for the confusion matrix is derived through observation of class c_k with $k \in [0, n]$. The representation is based on [80, p. 71].

- The **TN** samples are correctly classified $c_p = c_l$, but not the observed class $c_k \neq c_p$.

So, for a multi-class problem with observed class c_k the resulting confusion matrix is given in [Figure 2.15](#).

Accuracy

The most basic metric function for the classification task is the *accuracy*, which calculates the percentage of samples that are classified correctly. Based on the terminology of the confusion matrix, the accuracy is defined by

$$\text{accuracy}(c_k) = \frac{TP_{c_k} + TN_{c_k}}{TP_{c_k} + TN_{c_k} + FP_{c_k} + FN_{c_k}} \quad (2.42)$$

for class c_k . This accuracy definition provides a measure of correctly classified samples for each class and is in the best case 1 and in the worst case 0.

Intersection over Union

The **IoU** is defined as

$$\text{IoU}(c_k) = \frac{TP_{c_k}}{TP_{c_k} + FP_{c_k} + FN_{c_k}} \quad (2.43)$$

which was introduced in the Pascal Visual Object Classes Challenge [81]. The **TN** are therefore not taken into account, which makes the **IoU** metric more reliable than the accuracy, especially in the case of unequal class distributions. On this basis, the **mean intersection over union (mIoU)** is defined as a measure for all n classes by averaging the single class **IoU**

$$\text{mIoU} = \frac{1}{n} \sum_{i=0}^{n-1} \text{IoU}(c_i). \quad (2.44)$$

In this thesis, **IoU** and **mIoU** are primarily applied to compare classification or segmentation approaches, since this metric is more reliable than accuracy and represents the state of the art [81–84].

3 The State of the Art

This chapter describes the current state-of-the-art of **LiDAR** sensors in adverse weather conditions. [Section 3.1](#) covers weather condition monitoring techniques on roads with stationary and non-stationary systems. [Section 3.2](#) looks at climatic influence on **LiDAR** sensors. [Section 3.3](#) describes the current state of the art when it comes to the elimination of weather effects in point clouds, a process also known as "de-noising". [Section 3.4](#) lists related work on publicly available datasets for the purpose of determining weather influence on **LiDAR** sensors. Finally, [Section 3.5](#) summarizes the current state-of-the-art and highlights various areas of research complemented by this work.

Parts of this chapter were previously published in [\[85, 86\]](#) and [\[87\]](#).¹

3.1 Weather Condition Monitoring

This section contains information about instrumental measurement systems for the **MOR** and precipitation intensity. We will be focusing here on condition monitoring systems that can be used in the automotive field - that is, systems that are mounted along the roadside or on-board sensors that are already integrated in state-of-the-art passenger cars or that can potentially be integrated in the near future. Conventional passenger cars have so far not been equipped with sensors for this purpose, with the exception of temperature sensors and sensors for qualitatively estimating rainfall and brightness so that the windshield wipers and headlights can be adjusted accordingly.

¹ Editorial modifications have been made to improve readability and to ensure consistency of nomenclature.

3.1.1 Measurement of Meteorological Optical Range

The World Meteorological Organization divides systems for measuring the MOR into two categories [47]:

- Systems that measure the extinction coefficient or transmission factor
- Systems that measure the scattering coefficient or intensity of light scattered

Measuring the Extinction Coefficient

In principle, the extinction coefficient can be measured with telephotometric instruments, visual extinction meters or transmissometers. Since transmissometers are the most commonly used devices, they will be explained in more detail in the following [47]. The mean extinction coefficient and MOR can be calculated by measuring the mean transmission between a transmitter and receiver in a horizontal cylinder of air. With this measurement principle, the transmissometer is closely related to the definition of the MOR (Section 2.3.1) and so yields accurate information on the true optical range. The typical accuracy of transmissometers is around 10% when the MOR is up to 60 times the baseline length [47]. This type of measuring instrument is designed for stationary use only, because the influence of the air stream affects the measuring distance, which can lead to unreliable results. This means that they cannot be used in the automotive sector, for example, where, in addition to the air stream, localized temperature differences on the vehicle prevent the occurrence of fog in the vicinity of the vehicle, meaning that it cannot therefore be measured.

An application example of this measurement technique is the fog warning system on the A8 in Ulm, Germany, which alerts drivers to the presence of fog and poor visibility by changing the variable message signs according to current visibility levels [88, 89]. This application of a transmissometer enables very detailed visual ranges with ± 5 m below 50 m visual range and ± 5 % above 5.000 m [90]. The visual range is derived on the basis of a local transmission measurement with a wavelength of 640 – 660 nm over a distance of around 20 cm [90]. The visibility is estimated on the assumption that a reduction in transmission usually occurs due to fog [89, 90]. A disadvantage of such systems

is interference by insects or other objects, which can influence the measurement path and lead to faulty measurements. In June 2020, for example, a fog warning was issued on a bright and sunny day due to the presence of insects along the measurement path [91]. At the time of writing, no solution to these kinds of false alerts has been found [91].

Measuring the Scattering Coefficient

The reduction of visibility in rain or fog is caused mainly by the scattering of water droplets and only to a minor extent by absorption². The MOR can therefore be measured by determining the scattering alone, not taking into account absorption [47].

The scattering coefficient can be determined concentrating a beam of light in a small volume of air and using photometric methods to determine the proportion of backscattered light. When interfering light sources are modulated, this method can be used during the day and at night [47].

Measuring the intensity of scattered light yields less precise MOR measurements than transmissometers. Due to the relatively low contamination of optical surfaces and their ability to measure the MOR over a wider range than transmissometers, however, an increasing number of these instruments are used in automated meteorological observing systems (e.g. visibility measurements on highways or to determine the presence of fog) [47]. One example of such a sensor is the *optical fog sensor* from [92]. Such measurement systems typically achieve an accuracy of around 10%, but are less accurate than transmissometers at low values of the MOR and are more affected by precipitation [47].

LiDAR sensors can also be used to detect the presence of fog. Their range of visibility is very limited but sufficient to detect the presence of fog [47].

² Note: The absorption influence increase in the presence of air pollution, dust or ice crystals.

3.1.2 Measurement of Precipitation

In weather forecasts, rain is classified by duration over time and referred to as showers, intermittent precipitation or continuous precipitation. The intensity of precipitation is typically determined by optical disdrometer systems, Doppler radar systems or rain gages [47].

Optical disdrometer systems use the depression of a horizontal light beam to detect falling particles of liquid or solid water in the atmosphere. The intensity of the precipitation can be determined on the basis of the drop in intensity on the receiver side due to the falling particles. The amplitude of the reduction corresponds to the particle size and the duration to falling speed [47].

Doppler radar sensors emit a vertical beam that is reflected by falling liquid or solid water droplets in the atmosphere. The velocity and size of the particles can be measured by the Doppler shift. The precipitation type can also be determined by measuring fall speeds at different altitudes [47].

Conventional rain gauges are designed to measure the amount of precipitation over a predefined period by collecting the water that falls over a unit area with known proportions. Smaller devices also provide an indication of rain intensity [47].

3.1.3 Qualitative Measurement in Non-Stationary Systems

In the automotive field, non-stationary systems are typically mounted on the vehicle itself. As an example, state-of-the-art passenger cars are capable of adjusting the wiper speed to the prevailing weather conditions, so a dedicated sensor is typically mounted behind the interior mirror on the windshield [93–95]. These sensors are generally not designed to provide an accurate measurement of meteorological variables but instead a qualitative value for meeting defined application criteria (e.g. adjusting the wiper).

3.1.4 Qualitative Measurement with Non-Dedicated Sensors

The description of non-dedicated weather monitoring systems focuses on the utilization of passenger car on-board sensors - for example, sensors that are designed specifically for environment perception in semi-autonomous or fully autonomous driving tasks and for the task of monitoring weather conditions. Any perception sensor that is influenced by adverse weather could essentially be used for weather condition monitoring, provided that the weather impact is quantifiable. For the purpose of environment perception for an autonomous vehicle, the sensor and perception algorithms are usually designed to be as robust as possible in the face of adverse weather conditions. For the recognition of the environmental conditions, however, the influence is obviously an important aspect. This section therefore evaluates the application of camera and **LiDAR** sensors for weather detection in passenger cars.

Camera-based Weather Detection

In 1998, Busch and Debers presented an approach for estimating fog visibility based on stationary camera systems mounted above highways facing the roadway [96]. For the visibility estimation, only the image area of a roadway is considered to be a region of interest, and the intensity gradient is derived for this area [96]. A gradient threshold value is used to determine the maximum visibility in image coordinates, and a visibility level in world coordinates is derived on the basis of the known camera position and a flat world assumption [96]. The threshold value of the intensity gradient was determined in experiments on the basis of images taken at known viewing distances. The resulting visibility estimate is given in multiples of 50 m above 300 m visibility and in multiples of 10 m below 300 m visibility [96]. This principle is not easily applicable, for non-stationary cameras because the assumptions based on the static camera position are not satisfied.

A non-stationary approach for a percentage visual range reduction based on camera images is presented by Pormeleau in 1997 [97]. The idea is based on a contrast degradation along similar roadway features like lane markings. The contrast degradation is determined on the basis of geometric transformation of an adaptive trapezoidal area into a bird's eye view. The approach does not take

into account the root cause of the contrast reduction and so cannot differentiate between rain, fog or other weather influences. The algorithm is evaluated on the basis of simulated fog from 100 to 700 m visibility. To obtain reliable results, the author recommends analyzing at least 30 sequential frames, due to the standard deviation of individual measurements between 0.2 and 0.3 [97].

In 2004, Hautiere et al. derived the visibility by estimating the extinction coefficient based on camera images [98]. The correlation between the luminance of the background or objects and the measured pixel values of the camera image is assumed to be linear. The region of interest for deriving the intensity is selected to be as homogeneous as possible and extend from the road directly in front of the vehicle to the sky. Estimated meteorological visibility is demonstrated using a sample image under real fog conditions without any ground truth information. Based on synthetic image data, the authors suggest good accuracy for low visibilities ($< 150\text{ m}$), but do not cite any specific numbers. Consequently, no statistical validation of the visibility estimation is available.

In 2008, Roser and Moosmann presented a classification-based approach for detecting different rain levels [99]. The approach is capable of distinguishing between light rain, heavy rain and clear conditions based on a single input image. Histogram features based on brightness or contrast are calculated over different region of interests of the input image and concatenated to a single input vector for a linear SVM. The error rate for the distinction between heavy rain and clear conditions is less than 1 %. For the three-class-case, however, the error rate is between 2.0 % for highway environments and as much as 14.8 % for rural and urban environments.

In 2013, Pavlic et al. introduced an algorithm for differentiating between fog and the absence of fog based on linear discriminant analysis (LDA) [100]. The grayscale input image of a conventional passenger car camera is transformed, normalized and processed by image processing filters (e.g. Gabor filter) into a feature vector. These features are selected with a PCA. The final classification is derived based on a LDA which results in an accuracy of 95.35 %. The image descriptor presented by Roser et al. is applied as a baseline for the fog dataset and results in an accuracy of 92.20 % for the same dataset and classification task.

Weather classification based on single camera images with CNNs are presented by Elhoseiny et al. This method involves fine-tuning the AlexNet [74]

for differentiating between sunny and cloudy images [101]. The proposed CNN architecture achieves 82.2% normalized accuracy and outperforms the baseline classifier (53.1%), which was introduced with the dataset itself by Lu et al. [102]. The dataset contains 10k outdoor images with a wide variety of viewpoints, landscapes, scenes, and locations, which makes the process of distinguishing between sunny and cloudy conditions a highly challenging classification task. Further improvements are introduced by Lu et al. with a collaborative learning approach, which combines hand crafted features with a CNN and achieves an overall accuracy of 91.4% [103].

LiDAR based Weather Detection

To the best of the author's knowledge, no publicly available approaches to classifying weather conditions based only on sparse point clouds captured with LiDAR sensors have been implemented, evaluated or published in the automotive field. The basic concept behind detecting weather conditions using vehicle on-board sensors - including the utilization of LiDAR sensors - is presented in [47] and [104]. However, no implementations, evaluations or test results are available.

3.2 Influence of Weather on LiDAR Sensors

The influences of weather on LiDAR sensors is a large and highly active field of research [105–117], although publicly available datasets including LiDAR sensors are commonly recorded under favorable weather conditions (e.g. [82, 118]). This section summarizes the current state-of-the-art regarding the influence of weather on LiDAR sensors.

3.2.1 Influence of Fog on LiDAR Sensors

Hasirlioglu et al. proposed a theoretical model using multiple reflections of rain droplets or fog to determine the influence of fog and rain for automotive perception sensors [110]. The principle of the model is based on a longitudinal layer representation. Within each layer, reflection, transmission and absorption

could occur. This approach uses multiple reflections and is verified by means of an experimental setup with a fog and rain simulator [108, 109]. The system developed has an overall distance of 10 m between the sensor and target, whereby the effective length over which the weather affects the LiDAR signal is 4 m. With this theoretical model and experimental setup, the influence of weather on LiDAR sensors is studied by analyzing the influence of rain and fog on the perception capability of a standardized Euro NCAP Vehicle Target (EVT) [119]. However, the target is optimized to represent a vehicle for visual camera sensors and not for LiDAR sensors. The results of the test setup show that radar sensors are generally more robust against fog than LiDAR and camera sensors, which are strongly affected by fog [108]. Since the fog density (e.g. meteorological visibility) is not quantified, it is not possible to directly ascertain the real-world environmental conditions.

The LiDAR dataset in dense fog by [117] enables a conclusion to real-world scenarios, because the dataset was recorded in a dedicated climate chamber [62] with a closed-loop-controlled visibility range. Note that the datasets used in this work were recorded in the same climate chamber in collaboration with the authors of [117]. In addition, actual objects are used to represent typical scenes in the road environment and to ensure a correct reflection behavior for the backscattered light. The detailed analysis by Bijelic et al. takes into account the total number of scan points in a single frame, the intensity, the maximum detection distance and sensor parameter tuning. According to [117], the detection range of state-of-the-art LiDAR sensors deteriorates at visibility levels of less than 40 m visibility and so is limited to 25 m, even with multiple returns.

3.2.2 Influence of Rain on LiDAR Sensors

The influence of rain on LiDAR sensors was analyzed in [107, 113]. Filgueira et al. presented a work that quantifies the influence of rain on one LiDAR sensor and a static scene, detailing the average range, intensity and number of points for certain objects [107]. The results show smaller changes in the distance of detected objects, while the intensity and the number of points decrease dramatically.

Ryde et al. and Phillips et al. analyze the influence of dust in detail with one type of 2D laser scanner, similar to our approach of studying the influence of fog and rain [113, 116]. Smoke and rain are also examined in [113], but no information regarding the influence of fog is available because the chamber used was not capable of producing artificial fog. According to [116], the influence of dust on LiDAR sensors is systematic and predictable because the LiDAR measures the leading edge of a dust cloud, which with the LiDAR used, occurs from a transmission of about 70 %.

Wojtanowski et al. [120] presented a detailed analysis and discussion concerning range degradation with hypothetical LiDAR sensors with $0.9\ \mu\text{m}$ and $1.5\ \mu\text{m}$ under foggy and rainy conditions. In terms of attenuation only by fog, rain and wet surfaces, LiDAR sensors with a wavelength of 905 nm outperform sensors with 1550 nm. Since air humidity did not have any significantly impact on the sensor's performance, fog is the most suppressing factor [120].

Kutilla et al. [121, 122] recently analyzed the impact of severe weather conditions on LiDAR sensors at 905 nm and 1550 nm by evaluating the SNR of the backscattered light and through a quantitative comparison of the number of points per object. According to [122], the 1550 nm LiDAR sensor outperforms the 905 nm sensor in adverse weather conditions, due to the lower restrictions on emitted light power to achieve laser class 1.

3.3 Point Cloud De-Noising

As demonstrated in the previous section, adverse weather conditions have a huge impact on the perception capability of LiDAR sensors. Consequently, point cloud processing algorithms either have to deal with these influences or require preprocessing by filter algorithms. Nevertheless, only a few de-noising algorithms for *sparse* point cloud data obtained by LiDAR sensors have been developed or are currently publicly available ([115, 123]). We will therefore first address the current situation regarding the de-noising of *dense* point clouds obtained by depth cameras or stereo vision and then examine how these concepts can be applied to *sparse* point clouds. The current state of the art regarding *sparse* point cloud data will then be summarized.

3.3.1 Dense Point Cloud De-Noising

Previous work on 2D depth image de-noising is based primarily on dense depth information obtained by stereo vision and depth cameras (e.g. Intel RealSense, Microsoft Kinect, etc.), which is why the traditional algorithms developed over many years for camera image de-noising can be applied in a straightforward manner [124, 125]. These approaches can be split into three different categories: (1) spatial, (2) statistical and (3) segmentation-based methods.

Spatial smoothing filters (1) - e.g. the Gaussian low-pass filter - calculate a weighted average of pixel values in the vicinity, where the weight decreases with the distance to the observed pixel. Points are smoothed as the distance from the derived weight increases [126]. For de-noising 2D point cloud data corrupted by snow, these filter types have proven highly successful, as shown by [127] with a median filter. Since it assumes only small variations in the vicinity, however, this approach generally fails to preserve edges. The bilateral filter, introduced by [126] for gray and color images, is replacing traditional low-pass filtering by providing an edge-preserving smoothing filter for dense depth images [128].

Statistical filter methods (2) for dense point cloud de-noising are often based on maximum likelihood estimation [129] or Bayesian statistics [130]. By optimizing the decision regarding whether or not a points lies on a surface, these approaches smooth surfaces and remove minor sensor errors.

By applying a segmentation step before filtering, segment-based filters (3) are smooth only localized point cloud segments with identical labels. This means that corners and finer structures are better preserved. Region growing [131], a maximum a-posteriori estimator [125] or edge detection [132] is used for segmentation, while bilateral filters are used for smoothing localized segments.

LiDAR point clouds are significantly less dense than camera images, particularly at larger distances. The direct application of camera algorithms therefore does not typically achieve the desired result, as exemplified in [123] for a median filter applied to point cloud data. Since conventional **LiDAR** sensors have a resolution of tenths of a degree and a range of 200 to 300 meters, the density of the point cloud decreases significantly in the middle and far range.

A basic machine learning approach for de-noising dense point clouds corrupted by fog with a visibility of 2 m and 6 m is presented in [115]. By manually extracting features, a **kNN** and **SVM** are trained. The feature vector in particular is based on the eigenvalues of the covariance matrix of the Cartesian coordinates, which means that it is derived only if there are more than ten points in a 50 mm^3 cubic voxel. For a sparse **LiDAR** point cloud, this assumption is rarely satisfied.

3.3.2 Sparse Point Cloud De-Noising

Many approaches in the 3D domain are based on the spatial vicinity or statistical distributions of the point cloud [133], such as the statistical **outlier removal (SOR)** and **radius outlier removal (ROR)** filter. The SOR defines the vicinity of a point based on its mean distance to all k neighbors compared with a threshold derived by the global mean distance and standard deviation of all points. The ROR filter directly counts the number of neighbors within the radius r in order to decide whether or not a point is filtered. Charron et al. [123] recently showed that these filter types are not suited for the de-noising of sparse point clouds corrupted by snow. In response, the enhanced **dynamic radius outlier removal (DROR)** filter was introduced by [123], which increases the search radius r for neighboring points as the distance from the measured point increases. Since this approach takes into account the raw data structure of **LiDAR** sensors, which is less dense at far distances, a better performance could be achieved.

Nevertheless, these approaches are based on spatial vicinity and so discard single reflections without points in the neighborhood. As a result, points at greater distances are increasingly filtered, as shown in [123] for the **SOR**, **ROR** and even **DROR**. Valuable information for an autonomous vehicle - especially at higher speeds - is therefore discarded and the sensor's range is additionally limited by the filter.

3.4 Datasets

In 2009, the first dataset with radar, visual camera, infrared camera and LiDAR sensors was recorded in challenging environmental situations (dust, rain and smoke) by Peynot et al. [106]. According to the results, LiDAR sensors exhibit significantly more attenuation than radar sensors in challenging environmental conditions. With LiDAR sensors, it was also observed that objects could disappear behind the airborne dust. Based on the different attenuation of the two sensor concepts, an algorithm was developed to remove the dust reflections by filtering the laser data based on radar data. Most recent datasets are recorded under favorable weather conditions only or include scenes with adverse weather only to a minor extent. The most prominent datasets are the "Ford campus vision and LiDAR dataset" by [118] from 2011 and, in particular, the "KITTI dataset" by [82] from 2013.

During the period when this work was created, large datasets for the purpose of autonomous driving were publicly released: the "nuScenes dataset" by [26] in 2019, the "Waymo open dataset" by [134] in 2020 and the "PandaSet" Hesai and Scale AI [135] in 2021. These large-scale datasets contain a variety of traffic situations, objects, road types and detailed ground-truth annotations for the purpose of semantic segmentation or object recognition. The datasets also contain scenes with different weather conditions.

3.5 Discussion of the Current State of the Art

Given that LiDAR sensors are key for autonomous driving and robotics applications, they are currently being developed by numerous companies in a wide variety of designs. Nevertheless, LiDAR technology is still heavily challenged in adverse weather because the range measurements are highly impaired by fog, dust, snow, rain, pollution and smog [85, 116, 117, 122, 123, 136]. Such conditions - from the point of view of the autonomous system - result in erroneous measurements in the point cloud data due to the reception of backscattered light from water droplets (e.g. rain or fog) or arbitrary particles in the air (e.g. smog or dust).

For environment perception algorithms, these points are undesirable noise that need to be specifically addressed in order for the system to be capable of properly interpreting the scene. This is particularly relevant for algorithms that make direct use of the low-level geometry of a measured point cloud, e.g. the Stixel algorithm [137], where noisy input data inevitably results in noisy and false Stixel output data. CNN-based LiDAR perception algorithms might be better capable of handling such issues given their learning capacity, thereby reducing the need for explicit handling of noisy measurements. For the CNN to cope with increased noise, significantly more representative training data is required.

Moreover, most LiDAR perception algorithms involve more traditional bottom-up approaches for tasks such as object detection since they are usually implemented on the LiDAR sensors themselves, which offer very limited computing power. This has sparked a large body of research on algorithms to detect and handle noisy point cloud measurements in a pre-processing step before perception algorithms are applied.

3.5.1 Weather Impact and Classification

As the current state of the art demonstrates, the weather has a significant impact on LiDAR sensors. Heavy rain or dense fog in particular can severely affect performance [105, 107–109, 111–117, 121, 122, 138, 139]. It is therefore of vital importance to recognize and quantify the impact of current weather conditions on LiDAR performance in order to develop robust perception and, in turn, autonomous systems. Above all, for fusion and trajectory planning of autonomous cars, it is important to reliably classify current sensor performance for optimally weighing sensor modalities. It is therefore essential that the impact of various environmental conditions on specific sensor systems is evaluated. Furthermore, it is important that the weather impact is classified so that system limitations can be reliably identified. Some initial concepts for detecting weather conditions using vehicle on-board sensors - including the utilization of LiDAR sensors - are presented in [47] or [104]. To the best of the author's knowledge, however, no publicly available approaches, to classify weather conditions based only on sparse point clouds captured with LiDAR sensors have been implemented, evaluated or published in the automotive field.

3.5.2 Weather De-Noising

Many different 2D image anti-aliasing algorithms have been developed that focus on smoothing noisy surface points resulting from marginal sensor errors [125–128, 130–132]. De-noising algorithms in 3D space are often based on spatial features to discard noise points caused by rain or snow [123, 133]. Since these techniques discard points based on the absence of points in their vicinity, smaller objects at medium to large distances might be falsely suppressed and identified as noise. Furthermore, these filtering techniques are not designed to work at close range, especially when points occur in dense clouds caused by e.g. fog, spray or drizzle. In these conditions, filtering based on the local vicinity is not effective.

3.5.3 Dataset

Since the publication of the "KITTI dataset" [82] in 2013, numerous further datasets focusing on environment perception in autonomous driving have been published (e.g. [26, 134, 135]). The datasets and their ground truth annotations focus here on semantic segmentation or object recognition and naturally include numerous different weather conditions. None of them, however, provide for adverse weather conditions with detailed weather ground truth information containing different levels of fog or precipitation intensities. Furthermore, no data captured in controllable environments like climate chambers with detailed weather information containing fog visibilities or precipitation intensities is publicly available, even though this is essential for the in-depth analysis of the impact of weather on sensors and subsequent algorithms.

4 Concept and Method

This chapter describes the basic concept and methods underlying the approach to the challenges outlined in this work. The first part of this chapter ([Section 4.1](#)) describes the methods for analyzing the influence of weather as well as machine-learning-based classification and segmentation methods. The second part of this chapter ([Section 4.2](#)) covers data acquisition in adverse weather conditions, while the third part of this chapter ([Section 4.3](#)) covers data labeling and data augmentation to facilitate machine-learning-based methods.

4.1 Methods

To answer the scientific questions underlying this thesis, we apply different methods in order to

- (1) quantify the impact of rain and fog on [LiDAR](#) sensors;
- (2) analyze the ability of an algorithmic concept to derive the weather conditions based on point cloud data; and
- (3) evaluate a concept for point cloud de-noising in rain and fog.

These methods are outlined in the following subsections; parts of this section previously appeared in [\[85, 86\]](#) and [\[87\]](#)¹.

¹ Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

4.1.1 Methods for Quantifying the Influence of Weather

As the current literature already indicates, **LiDAR** sensors are heavily influenced by weather conditions [105–117]. It is therefore extremely important for autonomous systems to precisely determine this influence and to classify it at each time stamp. The quantification of influence is presented in the following sections, based first on the point cloud representation of individual objects and, second, on one complete point cloud containing the entire field of view or region of interest. The analysis for an entire point cloud could also be referred to as manual feature extraction, serving as a vector for machine learning methods, which are applied in the subsequent section.

Impact on Object Sensing

Object sensing is a crucial task for autonomous driving and the basis for all the following systems, including fusion, trajectory planning and behavior. Since **LiDAR** perception algorithms usually detect objects on the basis of geometric features of segments of the point cloud, two metrics are used for analyzing the alteration of those object segments at a constant distance.

We therefore analyze the point cloud representation for the object O as a function of rainfall rate R and visibility V at time stamp k . Thus, the number of points per object O and echo e is denoted by the function $N(O, e, k)$. The expected number of points N_{clear} for one object is based on the mean over all frames in favorable weather conditions without any fog or precipitation

$$\bar{N}_{clear}(O, e) = \frac{1}{K} \sum_{k=1}^K N_{clear}(O, e, k). \quad (4.1)$$

The number of frames or samples is denoted as $k \in 1 \dots K$. To assess the influence on perception performance, a point density PD is derived as a key metric, quantifying the impact of missing points. The density $PD(O, e, k)$ is defined for each return e and object O separately by calculating the ratio

of points per object $N_{weather}(O, e, k)$ scaled by the mean over all frames in favorable weather conditions \bar{N}_{clear} without any fog or precipitation:

$$PD(O, e, k) = \frac{N(O, e, k)}{\bar{N}_{clear}(O, e)} \quad (4.2)$$

The object density is an indication of the degradation of object perception and reflects the number of missed points on objects in adverse weather. In addition, the point density rate in clear conditions is an indication of the stability of the raw data and the occurrence of missing points (FP).

Impact on Point Cloud Data

Research into the impact of rain and fog on entire point clouds is based on the alteration of distance and intensity images in controlled environments with static scenes, whereby the metric to derive the influence on the overall point cloud is based on manual feature extraction based on statistical representations of the point cloud.

As explained in [Section 2.1.2](#), we utilize an unstructured point list as matrix $\mathbf{P}^{n \times m}(k)$ at a discrete time step k , where each row $i = (1, \dots, n)$ contains one point. The attributes of a point \mathbf{P}_i are defined as:

$$\mathbf{P}_i = (x_i, y_i, z_i, r_i, \theta_i, \varphi_i, e_i, \zeta_i). \quad (4.3)$$

The notation for one point is: (x, y, z) for the Cartesian coordinates, (r, θ, φ) for the spherical coordinates, e for the return number, ζ for the intensity or [EPW](#) depending on the sensor type. Manual feature extraction is based on the point list $\mathbf{P}(k)$ and starts with spatial filtering to focus on a **Region of Interest (ROI)**. Since the return energy of light scattered by atmospheric particles is weak, the influence of ambient conditions is expected to manifest itself primarily at close range. The point cloud is thus spatially filtered, restricting processing to the near-range ($x \leq 20m$) of the ego lane ($-1.5m \leq y \leq +1.5m$). Furthermore, the [ROI](#) reduces local dependencies because roadside vegetation, guardrails, etc. are filtered out, saving computation time and resources. At the same time, the [ROI](#) is sufficiently large to provide a representative segment of the entire point cloud. A distinction by the echo number $t \in \mathbb{N}$ representing the first, second or third return signal of a transmitted light pulse, is reasonable because the

[h]

Table 4.1: Feature vector for environment classification based on point cloud data. For the feature set "VLP16", the echo pulse width epw is replaced by the intensity I .

f_1	N_1	f_5	\bar{r}_2	f_9	$\text{mean}(r)$	f_{13}	$\text{mean}(epw)$
f_2	N_2	f_6	\bar{r}_3	f_{10}	$\text{mean}(\varphi)$	f_{14}	$\text{eig}(\text{cov}(x))$
f_3	N_3	f_7	$\text{mean}(e)$	f_{11}	$\text{mean}(\theta)$	f_{15}	$\text{eig}(\text{cov}(y))$
f_4	\bar{r}_1	f_8	$\text{var}(e)$	f_{12}	$\text{var}(epw)$	f_{16}	$\text{eig}(\text{cov}(z))$

number of echoes per scan direction relates not only to the number of objects but also atmospheric particles, which are potential scatter points for the light. Hence, the amount M_t is defined as $M_t := \{e_i | e_i = t\}$ with $t \in \{1, 2, 3, \dots\}$ being the number of the respective received return pulse per angle. For the number of points for a specific echo, the signal $N_t(k)$ is derived:

$$N_t(k) = |e_i(k)| \quad \forall e_i \in M_t . \quad (4.4)$$

The mean and variance of one attribute p_j are calculated for each frame by:

$$\bar{p}_j(k) = \frac{1}{n} \sum_{i=1}^n p_{ij}(k) \quad \text{var}(p_j(k)) = \frac{1}{n} \sum_{i=1}^n (p_{ij} - \bar{p}_j(k))^2 . \quad (4.5)$$

For example, the mean distance of all points corresponding to a specific return is given by:

$$\bar{r}_t(k) = \bar{r}_i(k) \quad \forall e_i \in M_t . \quad (4.6)$$

The spatial distribution of the points is represented by the eigenvalues of the covariance matrices of x , y and z , similar to [115]. Finally, the assignment of the resulting feature vector $f = (f_1, \dots, f_{16})^T$, shown in Table 4.1, describes one frame of the laser scanner. The features are filtered in a subsequent component analysis to identify the parameters with the highest effect [140]. Additionally, the impact of a static scene has to be mitigated in order to not bias the training of ambient condition detection. For example, the total number of points is not taken into consideration for weather classification because it is highly dependent on the situation (empty highway versus inner-city traffic jam).

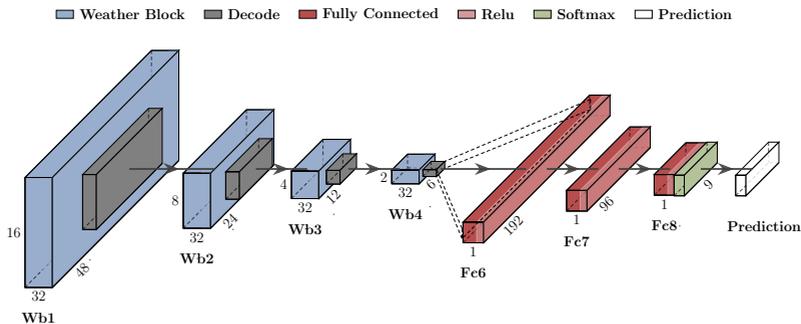


Figure 4.1: Basic *CNN* architecture for weather classification based on **LiDAR** point clouds. The denoted WeatherBlock (WB) shows an inception module, which is followed by a decoding layer based on a convolution.

4.1.2 Methods for Weather Detection by LiDAR Sensors

The most successful state-of-the-art camera image weather classification approaches are based on machine learning, either with manual feature extraction [99, 103, 141, 142] or deep learning approaches [143, 144], as presented in Section 3.1.4. We therefore apply these methods to the task of weather classification by means of **LiDAR** point clouds.

Specifically, the manual extracted features presented in Section 4.1.1 are derived to quantify the weather impact and are therefore suitable for use as a feature vector for an **SVM** and **kNN** classifier.

Additionally, deep learning techniques with a **CNN** are directly applied to the raw point cloud as an end-to-end approach to classifying the weather conditions based on the sparse point cloud data. The development process of the applied **CNN** architectures is explained below. For framewise weather classification, we adapt state-of-the-art **CNN** architectures for sparse point cloud semantic segmentation for the classification task [145, 146]. Since these architectures are capable of retrieving information at a point level and predicting a label per point, we use the backbone structure to create a new network architecture for classifying the weather condition for the entire image.

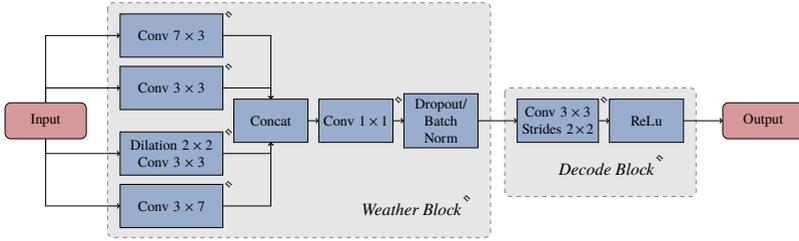


Figure 4.2: Details of the WeatherBlock and DecodeBlock structure. The WeatherBlock is presented in [87], based on the LiLaBlock from [145] and enlarged by a dilated convolution [129] and a dropout or batch normalization layer. For the classification task, down-sampling is additionally applied after each WeatherBlock by a convolution with strides 2×2 . A ReLu layer is used as an activation function [76].

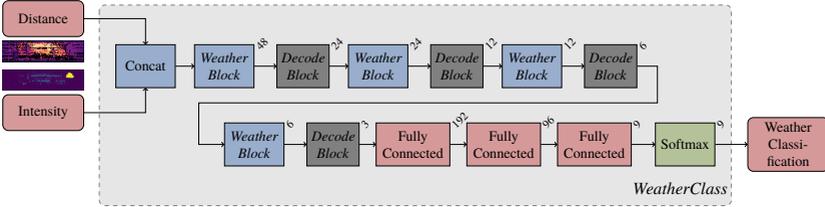


Figure 4.3: Block diagram of the *WeatherClass* architecture for end-to-end weather classification based on LiDAR point clouds.

As input data, we use the LiDAR distance and intensity image (see Section 2.1.2). If the sensor data contains multiple returns, we concatenate the strongest and last return of the intensity and distance image. The data is then processed by a CNN in order to predict the weather condition.

The model architecture is based on a combination of *LiLaNet* from [145] and *RangeNet* from [146]. Figure 4.1 illustrates the architecture of the classification CNN, whereby the input shape, number of output predictions, depth and number of layers are flexible. These parameters are adapted to the complexity of each classification task, for example, to the number of classes or the fineness of the subdivision of weather conditions. The WeatherBlock (Figure 4.1) is an inception module with four parallel convolutions, a bottleneck convolution and dropout or batch normalization layer. Figure 4.2 shows the WeatherBlock in detail, which is inspired by the *LiLaBlock* introduced by [145]. Compared

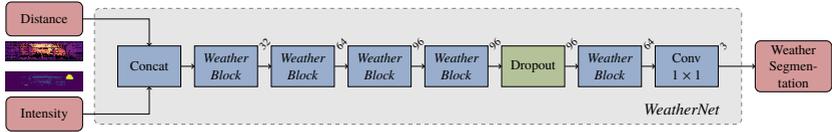


Figure 4.4: The proposed *WeatherNet* architecture is based on *LiLaNet* presented by [145] and optimized for de-noising purposes. The depth is therefore reduced, a dropout layer is inserted and a dilated convolution is added to the base block of the network. The modified *LiLaBlock* is shown in detail in Figure 4.2. The *WeatherNet* illustration previously appeared in [87].

with [145], we enlarged the *LiLaBlock* with a squared dilated convolution to provide more information about the spatial vicinity by increasing the receptive field. Furthermore, a dropout or batch normalization layer is applied to improve generalization capability [147, 148].

Each *WeatherBlock* is followed by a down-sampling layer that has been adjusted based on [146]. We used a 3×3 convolution with 2×2 strides, followed by a leaky ReLU activation function by [76]. The input shape is therefore divided in two vertically and horizontally. Subsequently, fully connected layers are applied until the target vector shape is achieved and the prediction is derived by a softmax activation layer. Figure 4.3 illustrates the aforementioned architecture for weather classification in the form of a schematic block diagram.

4.1.3 Methods for Point Cloud De-Noising

We propose a filter approach based on a convolutional neural network, which understands the underlying data structure and can generalize its characteristics for different distances and clutter distributions. Furthermore, this approach can incorporate the intensity information of the point cloud. The semantic segmentation task is being further developed by many in the scientific community and is already applied to the *LiDAR* point cloud domain, showing very promising results [145, 146, 149]. A major advantage is that the algorithms can generalize to a very high standard and thus recognize objects at different distances and orientations.

A number of approaches already exist for the input data layer and the network structure itself, which we utilize and adapt to the task of semantic weather

segmentation [145, 146, 150–153]. The recently introduced *PointPillars* by [152] is based on a feature extraction network, generating a pseudo image from the point cloud, which is then used as input for a backbone *CNN*. This approach excels with KITTI’s object detection challenge [82] in terms of detection performance and inference time. Just such a 3D *CNN* architecture was recently applied to a pointwise semantic segmentation task by Piewak et. al, as a further development of the aforementioned 2D approach. In terms of efficiency and computation complexity, however, 3D approaches still have a number of disadvantages compared with 2D approaches. Since preprocessing algorithms particular place strict requirements on computation speed, we will focus on 2D input layer approaches, which commonly use a bird’s eye view (BEV) [150–152] or an image projection view [145, 146, 153]. We therefore propose a 2D approach inspired by the *CNN* architecture of *LiLaNet* [145].

Specifically, for the de-noising of *LiDAR* images, we will utilize the *WeatherBlock* as introduced in the previous section and already published in [86] and [87]. Since labeling is performed on a single point level, the decoding layer, which reduces the image resolution, is omitted. Due to the reduced complexity of the segmentation task (3 classes) compared with multi-class semantic segmentation (e.g. 13 or 28 classes) [82–84, 145], we reduced the depth of the network by reducing the number of filters. A dropout layer is also added before the last *WeatherBlock*. The resulting network architecture is shown in [Figure 4.4](#).

4.2 Data Acquisition

To address the questions posed in this work through the application of machine learning methods as proposed in [Section 4.1](#), a large-scale dataset for *LiDAR* sensors containing various different weather conditions with detailed ground truth information, is required. The requirements for such a dataset were therefore first defined in order to evaluate whether existing datasets can be used or new data needs to be acquired. Furthermore, the requirements enable planning for the acquisition of future datasets.

4.2.1 Requirements Regarding Datasets

The requirements regarding the datasets are derived by the applications of this thesis: the analysis of weather influences on **LIDAR** sensors and the machine learning approaches for the classification and filtering of these influences. First, the repeatability of weather conditions is essential in order for the same scenarios to be recorded repeatedly under different weather conditions. The influence of weather can then be analyzed exclusively while the surroundings remain static. Second the predictability of weather conditions is necessary for capturing a significant amount of data from very rare weather conditions like dense fog [154]. In addition, either constant environmental conditions or very precise ground truth reference systems are required to capture an entire scene with known weather characteristics.

While a climate chamber fulfills the requirement for repeatability, it is almost impossible with road-based recordings to create an identical scene in different weather conditions or identical weather conditions in different scenes. Moreover, a climate chamber provides a range of opportunities and information for a recorded dataset such as highly detailed ground truth information about the weather condition, closed-loop controlled stabilized meteorological visibility, rainfall rate or temperature. A controlled environment is therefore used for recording adverse weather data.

Datasets recorded in controlled environments, however, are limited in terms of the variance of weather conditions, variety of scenarios and size of the available recording area. Datasets recorded on the road are important to assess the influence of natural weather and the performance of the algorithms under realistic conditions. This is why we combine on-road and climate-chamber-based recordings in order to leverage the benefits of both environments in one dataset. The requirements regarding a dataset suitable for the questions contained in this thesis are as follows:

Requirements for Controlled Environments

- REQ-01 The weather simulation area must be sufficiently large to present realistic scenarios for analyzing sensor perception capability.
- REQ-02 The simulated weather conditions must be verifiably equivalent to real-life weather conditions.
- REQ-03 The simulated weather conditions must be quantified using meteorological metrics and provided as ground truth information.
- REQ-04 Predictability: The controlled environment must be capable of generating rare weather conditions instantaneously.
- REQ-05 Repeatability: The controlled environment must be capable of generating reproducible weather conditions in order to capture different scenarios under the same weather conditions and the same scenario under different weather conditions.
- REQ-06 Controllability: The simulated weather conditions must be continuously controlled. The simulated weather remains constant during the measurement recording, or is continuously monitored by measurement technology.

Requirements for Road Environments

- REQ-07 Various natural rain conditions, including different precipitation intensities.
- REQ-08 Various natural fog conditions, including different densities in terms of MOR.
- REQ-09 Various traffic situations and road conditions.
- REQ-10 Road environments must allow algorithms trained in controlled environments to be evaluated in real-life traffic situations. The weather conditions therefore need to be measured or at least documented as scene tags in order to generate ground truth information. The frequency of this information is to be at least one tenth of the frame rate of the LiDAR sensor.

As explained in [Section 3.4](#), no dataset for autonomous driving with the focus on adverse weather is currently publicly available. Moreover, none of the

Table 4.2: Comparison of publicly available [LiDAR](#) datasets in automotive environments based on the derived requirements for this work.

	Requirement REQ-XX	Ford C. [118]	KITTI [82]	nuScenes [26]	Waymo [134]	PandaSet [135]	Ours [87, 155]
Controlled Environment	01 Realistic Traffic Scenarios	✗	✗	✗	✗	✗	✓
	02 Realistic Weather Simulation	✗	✗	✗	✗	✗	✓
	03 Groundtruth (meteorol. metrics)	✗	✗	✗	✗	✗	✓
	04 Predictability	✗	✗	✗	✗	✗	✓
	05 Repeatability	✗	✗	✗	✗	✗	✓
	06 Controllability	✗	✗	✗	✗	✗	✓
Road Environment	07 Natural Rain	✗	✗	✓	✓	✗	✓
	08 Natural Fog	✗	✗	✗	✗	✗	✓
	09 Diversity	✓	✓	✓	✓	✓	✓
	11 Groundtruth (Scene Tags)	✗	✗	✓	✗	✗	✓

published datasets meets the requirements summarized in [Table 4.2](#). In particular, no dataset acquired in a climate chamber and under controlled weather conditions is currently publicly available.

Multiple datasets will therefore be collected in this work with a focus on unfavorable weather conditions. The datasets focus primarily on the weather conditions "fog", "rain" and "snow". Other weather conditions were not explicitly considered but are included in the dataset as a result of the outdoor recordings. Most of the recordings were made in collaboration with the Dense project [156]. In February 2020, the recorded datasets were published as part of the following contributions [87, 155, 157–159].

4.2.2 LiDAR Sensors Used

Since the market for [LiDAR](#) sensors in the automotive sector is currently very dynamic, the performance of the sensors, - especially in terms of range and resolution - is constantly evolving. This means that any [LiDAR](#) sensor setup will likely be outdated as soon as a dataset is captured, pre-processed and published, which is why we have continuously updated the selection of our sensors with recently published state-of-the-art [LiDAR](#) sensors to ensure that we are using the most up-to-date sensor for the respective recording of our

Table 4.3: All sensor specifications are based on the manufacturer’s data sheet [20–24]. We stated only the specification of the configuration that we used during our recordings.

Manufacturer	Sensor Name	FoV		Resolution		Framerate	Number of Returns	Maximum Range
		h.	v.	h.	v.			
Velodyne	VLP16	360	30.0	0.2	2	10 Hz	2	100 m
Velodyne	HDL32E	360	41.33	0.08 – 0.33	1.33	10 Hz	2	100 m
Velodyne	VLP32c	360	40.0	0.1 – 0.4	0.33	10 Hz	2	200 m
Velodyne	HDL64E-S3	360	26.9	0.08 – 0.35	0.4	10 Hz	2	120 m
Ibeo	Scala	145	3.2	0.25	0.8	12.5 – 22.5 Hz	3	150 m

datasets. This is why our recordings involve numerous different sensors. Table 4.3 contains an overview of important information about the sensors used for data acquisition in adverse weather.

The sensors used for data acquisition in this work are four state-of-the-art Velodyne LiDAR sensors: Velodyne VLP16, Velodyne VLP32c, the Velodyne HDL64E-S3 and Ibeo Laserscanner Scala. "NuScenes" - the publicly available dataset used here - is captured with the Velodyne HDL32E [26]. All sensors operate at wavelength of around 905 nm with a scanning system. The main difference between the Velodyne and Ibeo sensor concepts lies in the mechanical design of the scanning pattern [85]. The Velodyne sensors use a rotator to spin the transmitter and receiver; the Scala sensor keeps the transmitter and receiver fixed and deflects sending and receiving light with a rotating mirror [85]. Another difference is that the Scala sensor provides the EPW of the received light pulses, whereas the Velodyne sensor measures the intensity of the received pulses [85]. Both sensors are capable of detecting multiple returns, which are also referred as "echoes". While the Scala sensor provides three echoes ordered by distance, the Velodyne sensor provides the last and the strongest echo. If the last and the strongest echoes are identical, the second strongest echo is provided [20–22, 24, 25]. Since the notation with the strongest and last return causes confusion if there are more than two returns, a uniform designation is used for the sensor types used, sorting the pulses by distance. We therefore refer to the returns as 1 for the closest return and the returns ascending with increasing distance. If we use the strongest or last echo, we state this explicitly. If no multiple reflections are detected, there are no points provided by the Scala sensor for echo 2 or 3. With Velodyne sensors, however, the strongest and the last echo are identical if only one return is measured.



Figure 4.5: Static setups in the chamber representing four complex, real-life traffic situations. The upper picture shows a scene with a pedestrian (1) on a zebra crossing (6), a child (7) imitating chasing a ball (4) on the street, a parked car (2), a cyclist pushing their bike across the street and a car (5) that is turning left. There is also a garbage can (8) on the far right. The bottom pictures show a number of traffic scenarios with different objects like a black tire as lost cargo, guardrails, cars, lane markings, reflector posts, traffic signs, a plant and pedestrian mannequins with and without umbrella [85].

4.2.3 Recorded Datasets

Controlled Environments

By taking part in the Dense project [156], the following datasets were acquired under controlled weather conditions in Cerema’s climate chamber [62, 64] in accordance with the publications of [87, 157–159]. The climatic chamber [62, 64], which provides spatial and temporal uniformity in addition to water droplet size distribution, is capable to determining in great detail the artificially generated weather conditions. The climate chamber therefore meets

all requirements that we derived for a dataset in controlled environments - with the exception of the requirement regarding the size of the test environment. The simulation area and height of the chamber are a disadvantage which is why our work focuses on the influence of rain and fog in the near range ($d < 30$ m). However, since our research focuses on dense fog (10 – 100 m) [37] and heavy rain (> 7.6 mm/h) [50], where the maximum visibility range of even state-of-the-art - as of 2019 - LiDAR sensors is very limited, the limited length of the climate chamber does not constitute a major drawback for the recordings. Nevertheless, the limited height has a stronger impact on the tests because the ceiling and nozzles are permanently visible in the LiDAR data.

To enable machine-learning approaches to data captured in controlled environments, we designed different setups for creating scenarios that were as realistic as possible within the limited size of the facility. First, only real objects were used instead of replicas like the Euro Vehicle Target (EVT) by the European New Car Assessment Program (NCAP) [119], which is designed to represent a passenger car for camera sensors but not for LiDAR sensors. The characteristic backscattering distribution of a real car is different to that of the EVT, which has, for example, no retro-reflective surfaces. By using real objects instead of replicas we cannot expect any differences regarding sensor perception in relation to objects compared with on-road recordings. This is a tremendous advantage when it comes to training algorithms in both domains. For static scenes only, pedestrian mannequins are used instead of actual people in order to ensure the exact same scene without any changes in position or gesture during the entire recording process under different weather conditions.

To analyze the impact only of rain and fog, highly realistic static scenarios - as shown in Figure 4.5 - are recorded under reference conditions without any precipitation or fog, rain with a stabilized rainfall rate of 15 mm/h, 33 mm/h and 55 mm/h and fog with a close-loop controlled meteorological visibility between 5 m and 110 m. The chamber dataset with static scenes is denoted as *chamber32stat* and contains 72 800 samples for training, validation and testing. For reducing temporal correlations among samples collected in the climate chamber, each setup is used only in training, validation or test split (see Figure 4.5). This results in a data split of approximately (60% – 15% – 25%) for training, validation and testing.

In addition, dynamic scenes are recorded with a pedestrian, cyclist and car. A total of six different trajectories and complex combinations of those three

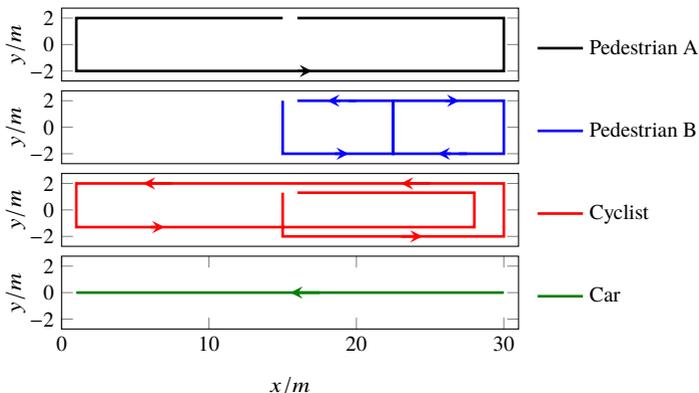


Figure 4.6: Schematic overview of the dynamic scenes in the climate chamber, recorded under different meteorological visibility ranges and rainfall rates. Overall, we obtain six different dynamic scenes: the four trajectories shown in the figure, the combination of a pedestrian A and the cyclist and the combination of all three objects. The climate chamber is 5.50 m wide and 31.0 m long [64].

dynamic objects are captured under each weather condition: the three rainfall rates mentioned above and under foggy conditions with constant visibility from 10–100 m in increments of 10 m. The dynamic scenes dataset (*chamber32dyn*) in the climate chamber contains a total of 150 790 frames. An overview of the trajectories is provided in [Figure 4.6](#)

Uncontrolled Environments

A large-scale road dataset under various fog, rain and snow conditions combined with favorable weather conditions was recorded by [157] and [155] in the Dense project [156]. We denoted this road dataset as *dense32* (VLP32c) and *dense64* (HDL64E-S3) for the different sensor types.

We also recorded a road dataset with a Velodyne VLP16 and a reference camera under various different weather conditions and traffic scenarios. The dataset contains sunny, cloudy and rainy weather situations during the day and the night. Furthermore, the recordings were captured on different types of roads (highways, rural- and urban roads) and in different traffic situations (empty

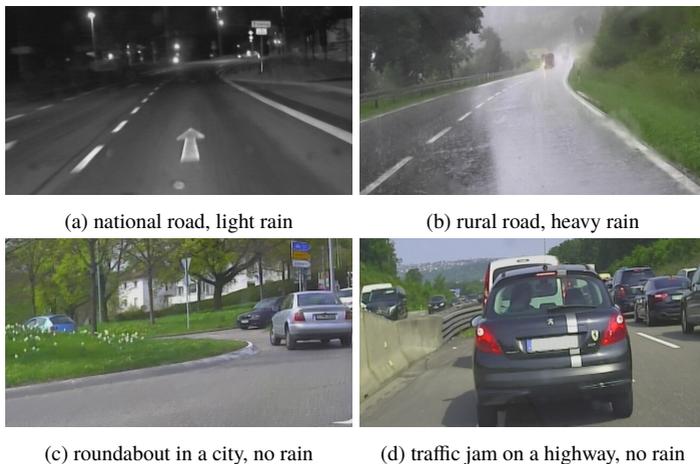


Figure 4.7: Examples for the road data set denoted as *rainkles16*, which contains different precipitation levels, scenarios and road types during day- and night-time [85].

roads, traffic jams, crowded scenes with numerous different objects). Figure 4.7 provides four different examples. The dataset contains a total of five scenes without precipitation (four during the day, one at night), three with occasional precipitation and four with almost permanent precipitation (three during the day, one at night). The dataset is referred to as *rainkles16* in the following.

We also use the recently released *NuScenes* dataset by [26], which contains a total of 331,036 LiDAR frames 64,718 of which were captured during rainfall. The sensor used here is the Velodyne HDL32E [21, 26].

Data Sets Overview

The notation in Table 4.4 uniquely identifies the datasets being used in this work. Furthermore, an overview of the number of samples and the purpose of each dataset is given. The datasets are recorded with different LiDAR sensors, which also enables the impact of adverse weather on different LiDAR sensor types to be analyzed. For deep learning purposes, however, the domain adaptation from one sensor type to another is still an open question in research [160] and not covered in this work. The classification models are thus trained on

Table 4.4: Overview of datasets used. The notation uniquely identifies the recording during the thesis. Except for NuScenes [26], all datasets were recorded within the scope of this work or in collaboration with the research project DENSE 24/7 [156].

Data Set Name	Sensors	Location	Purpose	Samples
chambers16	VLP16/Scala			274k/105k
chambers32stat	VLP32c	chamber	dense fog and heavy rain	73k
chambers32dyn	VLP32c			151k
rainkles16	VLP16		adverse weather (rain)	270k
dense32 [155]	HDL64 S3D, VLP32c	roads	adverse weather (snow, rain and fog)	1,400k
NuScenes [26]	VLP32E		object detection	331k

each dataset separately.

4.2.4 Data Split

We carefully split our datasets into training, validation and test data in order to preventing machine learning approaches from over-fitting. Since adverse weather conditions usually occur less often than calm weather conditions, datasets recorded on roads are mostly imbalanced (e.g. [26, 155]). This is important in the training of machine learning classifiers, by either distributing the class equally or weighting the classification by the inverse of the number of classes. Appendix Figure A.3 provides an overview of the class distributions of all datasets.

Furthermore, the datasets contain sequential frames or sequences that form temporal correlations, which must be specially taken into account when machine learning is applied. A recorded sequence or setup in the climate chamber, therefore, is either used for training, validation or testing. The target distribution of (65% – 15% – 20%) for training, validation and testing could therefore not be achieved accurately (Figure A.3).

4.3 Ground Truth Labeling and Data Augmentation

Ground truth annotations are essential for evaluating the quality of machine learning approaches. Furthermore, for the applied supervised methods, large-scale annotated training data is required. In this thesis, we classify point clouds on a frame basis and on a point level. Each frame or single point in one frame thus needs to be annotated (e.g. rain or no precipitation).

4.3.1 Framewise Labeling

For framewise labeling, one label per frame is sufficient. We have defined the following major classes: "clear", "fog" and "rain". "Clear" indicates no precipitation, fog or other adverse weather conditions. "Rain" and "fog" indicate the respective weather conditions and are subdivided according to the classification task and available ground truth information.

In controlled environments, we have highly accurate information about the weather condition on a frame level, since the ground truth data obtained by the chamber's reference system can be directly applied for annotating each [LiDAR](#) frame. The frequency of the ground truth information is 1 Hz.

For the *rainkles16* road recordings, the ground truth labels are derived by using the vehicle on-board sensor for adjusting the speed of the wiper according to the rain intensity on the windshield. The [LiDAR](#) point clouds in the *dense32* and *dense64* datasets are manually annotated by evaluating time-synchronized camera images. In the *NuScenes* dataset, the scene description itself contains information on whether or not it is raining. This is applied directly as a label for all frames in the corresponding scene [26].

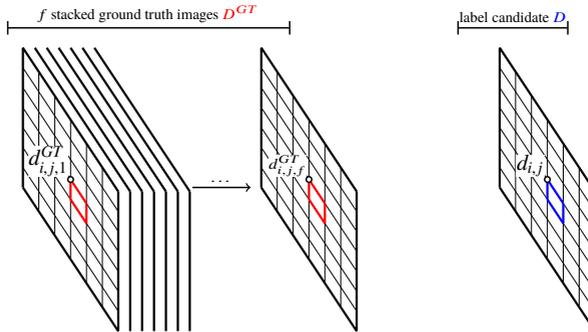


Figure 4.8: Diagram showing the automated labeling process for point clouds in static environments. The selected pixels $d_{i,j}$ are marked red for ground truth and blue for the frame that is labeled.

4.3.2 Pointwise Labeling

Parts of this section on the label method for pointwise annotations in reference, fog and rain conditions previously appeared in [86] and [87]².

For pointwise segmentation, ground truth labels are required for each single point. The label defines whether a point is caused by a water droplet or solid object. Since LiDAR sensors are capable of providing more than 1,200,000 points per second, manual annotations are extremely sophisticated.

For sparse LiDAR point clouds, manual annotation is also very challenging and even more difficult for semantic weather segmentation, whereby it is determined whether or not a point is caused by a water droplet. Humans are more readily able to process camera images than LiDAR point clouds, so a time-synchronized camera image as additional information is helpful for labeling LiDAR point clouds in order to significantly improve the label quality. Since water droplets cannot be captured directly by passive camera sensors, especially over long distances, this label aid is not available for semantic labeling of weather information. We therefore use the recorded static scenes in controlled environments to develop an automated labeling procedure, which

² Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

does not involve human perception. The basic idea of auto-labeling is that, in a static scene, every point that occurs in rain or fog in addition to the reference conditions is caused by the influence of weather and is labeled as "clutter". Each single LiDAR image k , therefore, is stacked to create l images under reference conditions. The stacked ground truth images are denoted as

$$\mathbf{D}^{\text{GT}} = (d_{i,j,k}^{\text{GT}}) \in \mathbb{R}^{(n \times m \times l)}. \quad (4.7)$$

Subsequently, as illustrated in Figure 4.8, each distance measurement d at pixel position (i, j) is labeled by calculating the difference to all stacked ground truth images $\mathbf{d}_{i,j,k}^{\text{GT}}$. Additionally, a threshold ΔR is added to the search region of valid distances. To minimize the number of false negatives, the threshold value $\Delta R = \pm 35$ cm is quite high in relation to the specified distance accuracy of the sensor. The labels $d_{i,j}$ - whether or not a distance measurement is "clutter" - are calculated per pixel as follows:

$$p = \begin{cases} \text{clutter,} & \text{if } \Delta R \leq \min_{1 \leq k \leq l} |d_{i,j,k}^{\text{GT}} - d_{i,j}| \\ \text{no clutter,} & \text{else} \end{cases} \quad (4.8)$$

$$(4.9)$$

By direct comparison of the pixel values, the transmitter and receiver components are identical in each comparison, which means that measurement offsets or any other absolute sensor errors have no influence. Furthermore, smaller sensor measurement inaccuracies are already taken into account in the comparison of the distance images through accumulation over multiple frames in the reference condition. This method is very fast because it does not require 3D information. Alternatively, a 3D point cloud comparison was implemented by a kd tree approach without showing significantly different label results. The sequences for fog and rain are labeled separately so that they can be differentiated.

The error of the derived ground truth labels is quantified by applying the label procedure to the reference recording itself. By splitting the reference frames into half, the evaluation is performed by taking the first split as ground truth labels to label the second one (and vice versa). Since the scenes are identical and there are no changes in weather conditions, all points are expected to be labeled as valid. For both tests, the resulting per-pixel mean false rate for labeling a valid point as clutter is 0.367 ± 0.053 %.

Since there are no manual annotated ground truth labels for clutter points, the error in relation to whether or not a clutter point is labeled as valid could not be evaluated.

4.3.3 Data Augmentation

The following section on the data augmentation techniques previously appeared in [86] and [87]³. State-of-the-art, publicly accessible sparse point cloud datasets tend to be recorded under favorable weather conditions. To use these datasets for semantic weather segmentation, we developed an augmentation approach for rain based on the fog model of [155]. We therefore obtained a large training dataset without requiring manual annotation while providing error-free ground truth. The augmentation algorithm is applied to LiDAR images to enable manipulations for each individual distance measurement, whereby the risk of occlusion is excluded by design. The proposed augmentation based on the model of [155] not only adds individual points but also alters additional attributes of the point cloud: Adverse weather affects the viewing range and also lowers the contrast of intensity and echo pulse widths.

Fog Model

First, the maximum sensing range is derived by the runway visual range

$$V = \frac{-\ln(C_T)}{\beta} \quad (4.10)$$

based on the atmospheric extinction coefficient β and the observer's contrast threshold C_T [155]. For LiDAR sensors, this threshold C_T can be interpreted as a detector threshold, whereby the sensor is capable of perceiving an object above the noise floor [155]. Since LiDAR is an active sensor system, the

³ Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

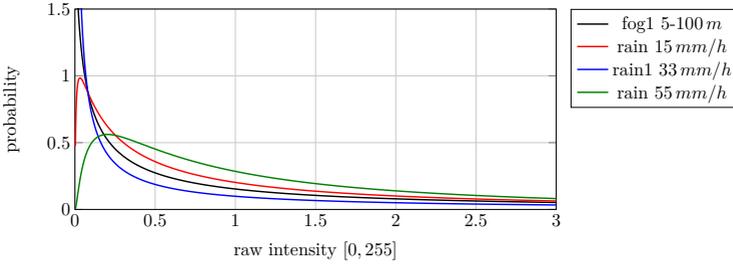


Figure 4.9: Intensity distribution of points whose existence can be explained by weather influences (e.g. rain or fog). The data is based on the climate chamber recordings at 5 – 100 m visibility and rainfall rates of 15 mm/h, 33 mm/h and 55. A logarithmic normal distribution $LN(\mu, \sigma^2)$ is assumed as the underlying probability density distribution function.

maximum sensor range is equal to half the maximum viewing distance and results in:

$$d_{max} = \frac{-\ln\left(\frac{n}{L_{fog}+g}\right)}{2 \cdot \beta} \quad (4.11)$$

The sensor threshold is a function of the received laser intensity L_{fog} , the adaptive laser gain g and the detectable noise floor n . Scattering points due to water droplets are added according to the model of [155]. In contrast to [155], the intensities of augmented points are derived from a logarithmic normal distribution $LN(\mu, \sigma^2)$, which is assumed as the underlying probability distribution function. Parameters μ and σ are derived from the intensities of all clutter points based on the auto-labeled climate chamber data from the previous section. We can therefore model the intensity distribution of fog at 10 – 100 m visibility and for rainfall rates of 15 mm/h, 33 mm/h and 55 mm/h. We preferred this method because in the model of [155], the original scene is mirrored in the intensity distribution because the augmented intensities \tilde{I} are a function of the perceived intensities I of the sensor ($\tilde{I} = I \cdot e^{-\beta \cdot d}$). The resulting logarithmic normal distributions are shown in Figure 4.9, whereby the rainfall rate of 33 mm/h corresponds more to the intensity distribution of fog than rain at 15 mm/h or 55 mm/h due to the technical characteristics of the climatic chamber.

Algorithm 1: Point cloud rain augmentation model $rain(D, I, \beta, R)$ with distance matrix D , intensity matrix I , atmospheric extinction coefficient β and point scatter rate p .

```

Function  $rain(D, I, \beta, p)$ 
   $B = \text{betafunction}(\beta)$ 
   $D_{max} = -\ln(\frac{n}{I+g}) / (2B)$ 
   $D_{rand} = \text{random.uniform}(D_{max})$ 
   $P_{lost} = 1 - \exp(-\beta \cdot D_{max})$ 
  foreach  $d \in D, d_m \in D_{max}, d_s \in D_{scatter}, d_r \in D_{rand}, p_l \in P_{lost}$  do
    if  $d_m < d$  then
      if  $p_l$  then
        pass;                                     ▶ point is lost, do nothing
      else if  $rand < p$  then
         $d = d_r;$                                ▶ random scatter point
         $i = LN(\mu, \sigma^2);$                    ▶ rain int.
      else
        pass
      else
         $i = i \cdot \exp(-\beta \cdot d);$            ▶ attenuate int.
      end
    end
  return  $D, I$ 

```

The augmented fog corresponds to a visibility V of 30 – 3000 m, which is why we use an atmospheric extinction coefficient β of between 0.001 and 0.1.

Rain Model

In addition to our modifications of fog augmentation based on [155], we optimized the rain augmentation system. The fog augmentation parameters have been adapted to make the augmented scatter points equivalent to natural rainfall. The atmospheric extinction coefficient β is set to 0.01 for rain augmentation. The point scatter rate p defines the per-point probability of random scatter points. The pointwise ground truth data obtained in this way enables the calculation of p for raindrops, which is 10.61 %, 0.73 % and 4.70 % for 15 mm/h, 33 mm/h and 55 mm/h in the climate chamber. The distributions of point-scatter rates are illustrated as a box plot in Figure 4.10. For the applied augmentation we finally fixed p at 7.5 %, which stabilizes the CNN training, matches the quantity of scatter points in natural rainfall and is within the range of the derived probabilities from the climate chamber. Furthermore,

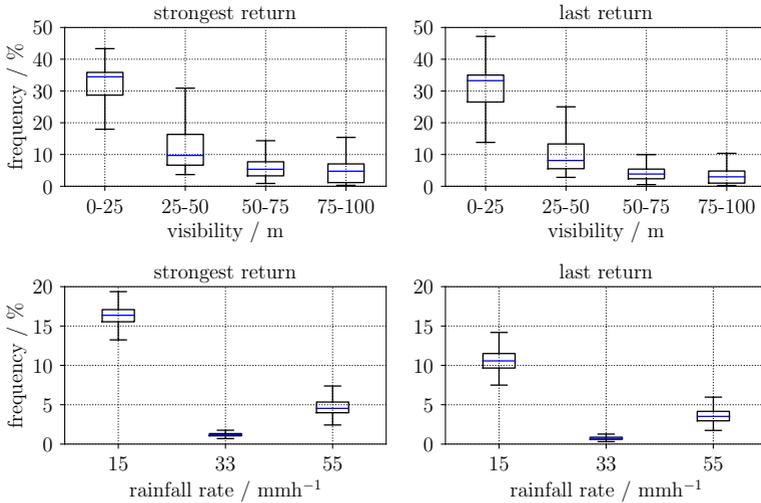


Figure 4.10: Illustration of the meteorological visibility in m and rainfall rate in mm/h provided by the climate chamber and the frequency for the number of scatter points during rainfall or fog. The frequency is derived by the ratio of the number of scatter points and the number of valid points.

the intensity distributions of the three different rainfall rates are used to sample the intensity of augmented scatter points caused by rain. In [Figure 4.9](#), the logarithmic normal distributions for 15 mm/h, 33 mm/h and 55 mm/h, as captured in the Cerema climate chamber [62], is given. The rain augmentation is described in detail in algorithm 1.

5 LiDAR in Adverse Weather Conditions

This chapter evaluates the aforementioned methods, which are applied to the data recorded during this work. [Section 5.1](#) provides an in-depth analysis of weather influence on [LiDAR](#) sensors. [Section 5.2](#) covers weather classification based on the aforementioned machine-learning approaches. [Section 5.3](#) covers the pointwise segmentation for sparse [LiDAR](#) point clouds corrupted by rain or fog.

5.1 Influence of Weather on Lidar Sensors

This section provides an in-depth analysis of weather influence on [LiDAR](#) sensors, focusing on the alternation of the point cloud data due to the impact of rain and fog.

5.1.1 Influence of Weather on Point Clouds

From the perspective of a [LiDAR](#) sensor, there are basically five different back-scattering behaviors that affect the point cloud data (see [Figure 5.1](#)):

- (a) Only returns from the cloud are received by the [LiDAR](#) sensor. Targets behind the cloud are not visible in the point cloud data.
- (b) The opposite behavior: Only the target behind the cloud is visible. The cloud is not present in the point cloud data.
- (c) A combination of (a) and (b) results in both, i.e. the cloud and target are present in the point cloud.

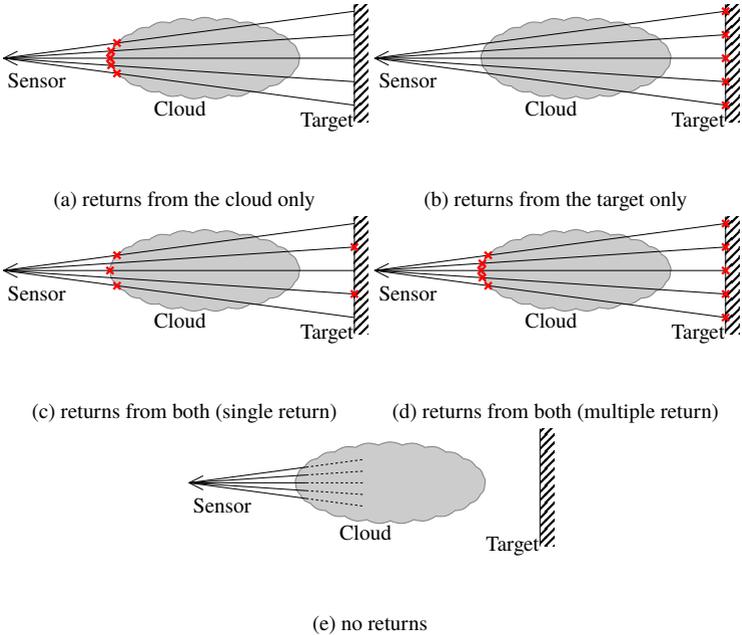


Figure 5.1: The backscattering behavior of LiDAR sensors in conjunction with particle clouds in the atmosphere, presented by [116] and including the possibility of multiple returns.

- (d) If the sensor is capable of detecting multiple returns, the cloud and the target can be perceived and are present in the point cloud for each receiver component.
- (e) The emitted light signal is absorbed by the cloud without backscattering enough energy to the receiver or transmitting enough energy to objects behind the cloud, which means that neither the cloud nor the object behind the cloud are visible in the point cloud data.

Weather Information available in LiDAR Data

This section was previously published in [86] and [87]¹.

Based on the dataset described in Section 4.2 and the corresponding ground truth labels described in Section 4.3, we can analyze the influence of weather at a point level (see Figure 5.2). The analysis shows that the LiDAR point cloud reflects the weather conditions in a great detail, since the number of points scattered by fog or rain is correlated with the visibility or rainfall rate. The results also indicate that no points are being lost (case (e) in Figure 5.1e) and so the sum of fog or rain and valid points is equivalent to the number of points in reference conditions.

The point cloud thus contains the information required for estimating the meteorological visibility or rainfall rate by determining the number of weather-induced scattering points. Since an increase in the rainfall rate does not necessarily result in an increase in scatter points, the rainfall rate cannot be estimated directly, but the extent of the degradation of the LiDAR sensor can be estimated. This information is incredibly valuable because it enables autonomous vehicles to adapt their behavior in line with the environmental conditions and sensor performance.

The low rainfall rate of 15 mm/h could be recreated in the rain simulation only by generating drizzle instead of normal rain as was the case with rainfall rates of 33 mm/h and 55 mm/h. The higher number of scattering points in the drizzle at 15 mm/h (Figure 5.2) compared with the higher rainfall rate of 33 mm/h shows that the sensor is affected more by drizzle at low rainfall rates than by rain at higher rainfall rates. This can also be observed in the point scatter rate shown in Figure 4.10.

5.1.2 Influence of Weather on Object Perception

Following analysis of the impact of weather on the entire point cloud, this section examines the impact on object sensing. On the basis of the static

¹ Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

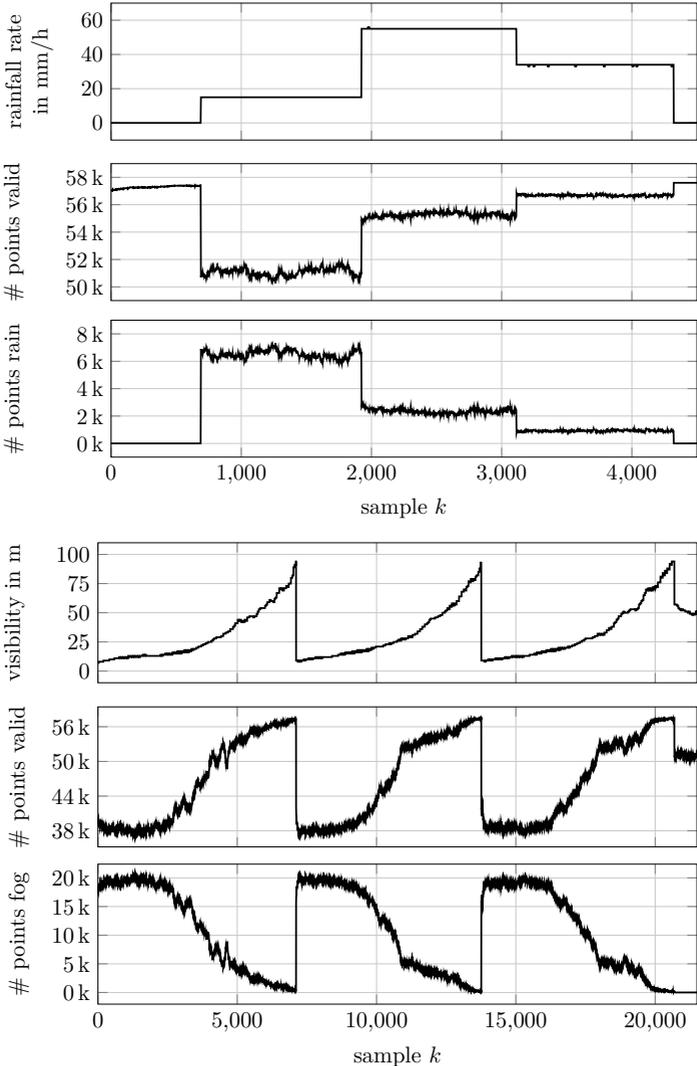


Figure 5.2: Illustration of the meteorological visibility in m and rainfall rate in mm/h provided by the climate chamber, the number of valid points and the number of scatter points during rainfall or fog [87].

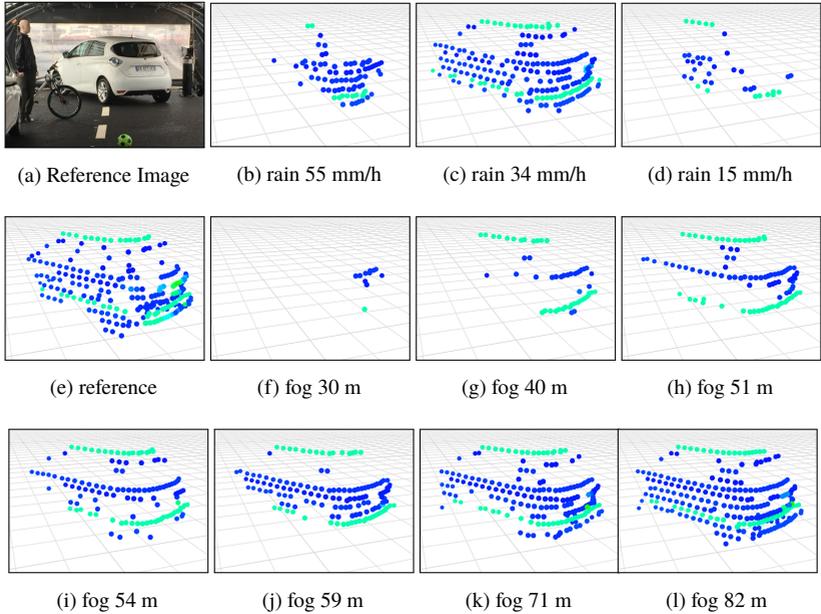


Figure 5.3: Influence of different fog and rain levels on point cloud data representation of a passenger car at a distance of 20 m, as captured by a LiDAR sensor. The corresponding dataset is 'chambers32stat', with its static scene 'setup 2' (Figure 4.5a).

scenes of the climate chamber dataset, we can evaluate sections of the point cloud for areas where prior known objects are located. The metrics mentioned in the Section 4.1.1 are applied. Parts of this section have already been published in [85]². The analysis is performed with the 'chambers16' and 'chambers32stat' datasets, which were introduced in Section 4.2.3.

Point Cloud Representation of a Car in Adverse Weather

Regarding the influence of fog and rain on LiDAR object sensing performance, we will illustrate some qualitative results directly on the point cloud first of Velodyne VLP 32c [22] and second of Scala [24].

² Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

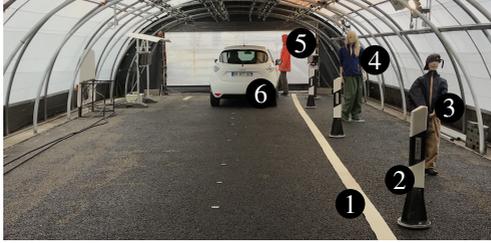


Figure 5.4: Experimental setup in the fog chamber for dynamic scenes in reference condition 'clear' without any rain or fog. The lane marking (1), reflector post (2), child and woman mannequin (3,4) and man mannequin with reflective vest (5) are stationary objects. The pedestrian, cyclist (both not shown) and car (6) are moving objects.

The representation of a car in the point cloud captured by Velodyne VLP 32C during different rainfall intensities and visibility ranges can be seen in [Figure 5.3](#). The car shown in the camera reference image ([Figure 5.3a](#)) is part of the static scene ([Figure 4.5a](#)) and located at a distance of 20 m from and at an angle of approx. 45 degree in front of the LiDAR sensor. As the reference camera image shows, the left side of the vehicle is partially occluded by a crossing pedestrian pushing a bicycle.

The first row of [Figure 5.3](#) shows the reference camera image ([5.3a](#)) and the point cloud captured during rain ([5.3b](#), [5.3c](#) and [5.3d](#)). In the second row, the point cloud under reference conditions can be seen in [5.3e](#), whereby the remaining figures ([5.3f](#), [5.3g](#), [5.3h](#), [5.3i](#), [5.3j](#), [5.3k](#), [5.3l](#)) are captured during increasing fog visibilities.

Comparing the reference scene ([5.3e](#)) with fog visibility above 71 m ([5.3k](#) and [5.3l](#)) and with a rainfall of 34 mm/h ([5.3d](#)), the representation of the car is nearly the same, except for the perception of low-reflective parts of the car like the tire, which is visible only in the reference scene.

In the rain scenes at 15 mm/h and 55 mm/h, the vehicle as a whole is no longer recognizable. So, for perception algorithms a classification as a vehicle is most likely no longer possible, although, it is still recognizable as an object of any shape. As already addressed in [Section 5.1.1](#), the influence of drizzle at 15 mm/h is higher than rain at 33 mm/h.

In foggy conditions where visibility is less than 60 m, the influence of the water droplets is significantly visible and the perception of the vehicle is considerably reduced. Below visibility of 50 m, the vehicle as a whole is no longer visible

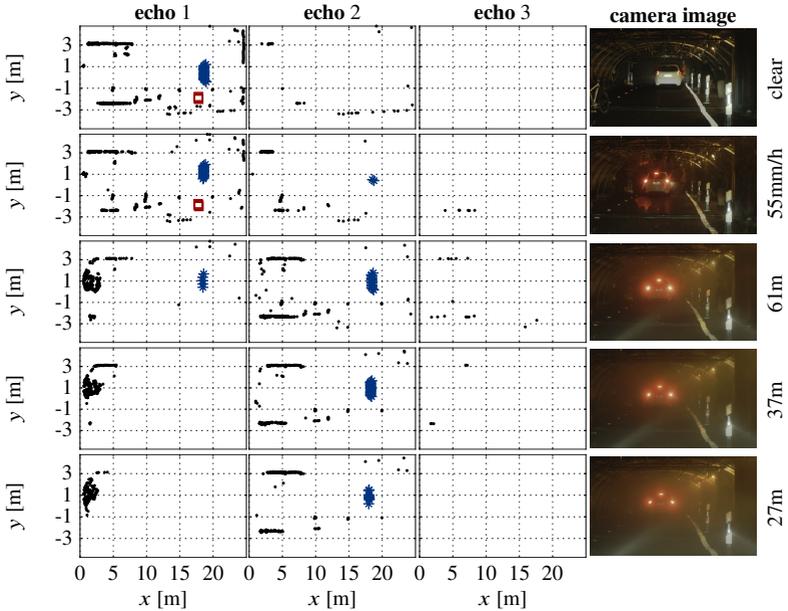


Figure 5.5: Bird's-eye view of the Scala sensor point cloud for a single random but representative frame k , split by echo number. An image from a state-of-the-art automotive camera is shown on the right; the meteorological visibility V is specified in meters on the far right in the case of fog and the rainfall rate in mm/h. For reference conditions, the label 'clear' is assigned. The pedestrian mannequin is highlighted with red boxes (object no. 4 in Figure 5.4) and the car in blue stars (object no. 6 in Figure 5.4). All other points are shown as black dots.

in the point cloud. Looking at the point cloud for visibility of 30 m, only individual points are still visible, but the vehicle as a whole is no longer recognizable. In this case, the visibility of the LiDAR sensor is less than the human eye, which can detect a vehicle at a distance of 20 m in fog with a visibility of 30 m.

In summary, the influence of fog on the LiDAR sensor under analysis here is greater than on human perception, since the object is already hardly recognizable at a visibility of 30 m. With decreasing visibility, low reflective surfaces are initially no longer perceived. As visibility decreases further, even the more-reflective surfaces can no longer be perceived, meaning that the object as a whole and the characteristic object shape are no longer recognizable. A

similar situation occurs in the case of rain. The evaluation is based on the perception of a real vehicle by a Velodyne VLP32C sensor in simulated weather conditions.

We then evaluate the bird's-eye view of the Scala sensor point cloud for a single random but representative frame k , split by echo number (see [Figure 5.5](#)). Additionally, an image from a state-of-the-art automotive camera sensor can be seen on the far right. The depicted point cloud is taken from 'chambers16' dataset with its setup shown in [Figure 5.4](#). The scene is the background setup for the recorded dynamic scenarios, which are referred as 'chambers32dyn'.

The point cloud with strong rain (55 mm/h) shows fewer points at the end of the climate chamber compared with the point cloud without any simulated weather. This is due to the reduced detection range of the sensor. The detection quality of objects such as cars is extremely important and interesting: For example, the car at a distance of around 19 m (highlighted in blue) is detected by both sensors in all scenarios, as shown in [Figure 5.6](#) and [Figure 5.7](#). In clear conditions, the first return is always received from the car; in fog and rain, however, the second echo contributes the majority to the detection of this car. The occurrence of second echoes from objects can therefore indicate for the presence of fog or rain.

In fog, within a visibility range of 50 m-60 m, a large number of first echoes is observed at a very short distance. Moreover, the detection quality and range is expected to be impaired because significant laser power is scattered by the atmospheric particles, leading to the other echoes. The environment perception capability and sensor range are limited. Only a few secondary echoes can be associated with fog because most coincide with the position of the car.

In dense fog (visibility of 20 m-40 m), the environment perception capability is severely restricted. Nearly all primary echoes are observed at a range of less than 5 m and are thus caused by fog. Nevertheless highly reflective targets like the retro-reflectors of the tail lights are still correlated with secondary or tertiary echoes. Comparing all fog and rain measurements with the clear ones, the number of second and third returns increases ([Figure 5.6](#) and [5.7](#)). Our evaluation also shows that a multi-echo sensor is beneficial because it also returns weaker reflections such as fog and rain while maintaining reasonable object detection performance compared with single-return sensors.

Comparing these results with the five range measurement behaviors of LiDAR sensors in the presence of dust, as presented in [116] and illustrated in [Figure 5.1](#), the influence of fog is similar to dust, in which the measuring range is the front of the dust cloud. The influence of rain, however, seems to be different.

Statistical Object Representation in Adverse Weather

For a statistical comparison of the object sensing performance, the accumulation of points that correspond to a car or pedestrian of at least 1, 200 frames for each weather condition are shown in a box plot in [Figure 5.6](#) for a Scala and in [Figure 5.7](#) for a Velodyne VLP16 sensor. The corresponding dataset is 'chambers16' with only static scenes; the car is a Renault Zoe and the pedestrian is a mannequin.

As shown in [Figure 5.6](#) and [Figure 5.7](#), the car and the pedestrian are perceived mainly by the first echoes under clear conditions. In rain, only some of the second and third echoes correspond to the objects. With decreasing visibility, the car and the pedestrian are perceived mainly by the second and even third echo in the case of the Scala sensor.

Furthermore, with decreasing visibility, the measured distance of the car decreases slightly, as does the number of outliers in terms of distance accuracy.

The variance of the [EPW](#) continuously decreases as the visibility range decreases. This observation holds true for the intensity measured by VLP16. Object detection algorithms that leverage intensity or [EPW](#) information are therefore likely to be strongly influenced by adverse weather conditions.

Finally, to access the influence on perception performance, a point density is calculated as a key metric, quantifying the impact of missing points. The metric used here is referred as object density and is presented in [Section 4.1.1](#). The density rate is based on the total number of points $N_t^O(k)$ from object O in frame k and scaled by the mean over all frames in reference conditions without any fog or precipitation. The object density is an indication of the degradation of the object perception. In [Figure 5.8](#) the resulting density is illustrated for a pedestrian and a car for both sensors. As a result, the perception of the car at 18 m remains quite robust during rainfall of 55 mm/h and degrades in fog

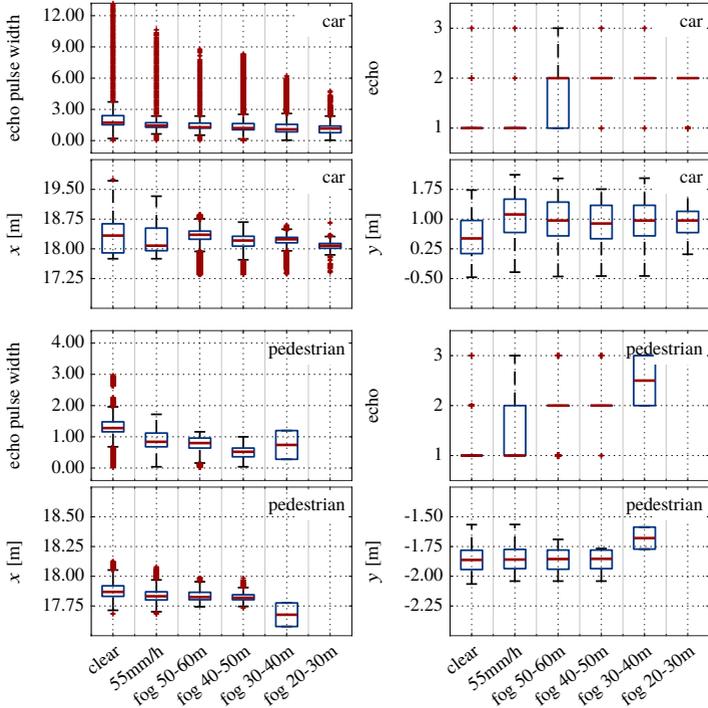


Figure 5.6: Object perception for the 'Scala' sensor with 1, 200 frames for each weather condition (except for rain, 921 frames). The box plot shows the result of the corresponding raw point cloud (x - and y -coordinate, echo and EPW) for a *car* and *pedestrian*. The weather is shown on the ordinate axes, ordered by descending meteorological visibility V . In case of fog, the visibility is stated in m; for rain the rainfall rate is stated in mm/h; the label 'clear' is used for reference conditions. The locations of the target objects are given in [Figure 5.5](#)

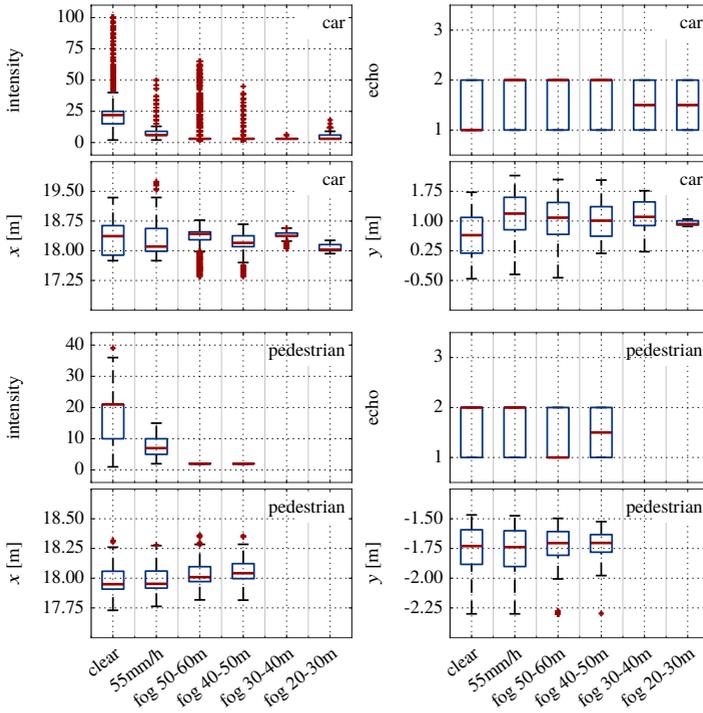


Figure 5.7: Object perception for the 'VLP16' sensor with 1,200 frames for each weather condition (except for rain: 921 frames). The structure of the figure is identical to Figure 5.6.

with a visibility range of 20 m–30 m to a median of 0.36 for VLP16 and to as low as 0.04 for Scala. In contrast, the detection density for the pedestrians at approximately 18 m declines significantly to 0.72 in rain for VLP16 but remains quite robust for the Scala sensor. In fog with a visibility range of 50 m–60 m, the pedestrian is mostly not detected by the VLP16. The Scala sensor is capable of detecting the pedestrian with a density of 0.87 down to a visibility of 50 m–60 m. Below a visibility of 40 m, the detection density for the pedestrian is 0. Consequently, objects without any retro-reflective materials are not perceived by LiDAR sensors in dense fog, even at close range.

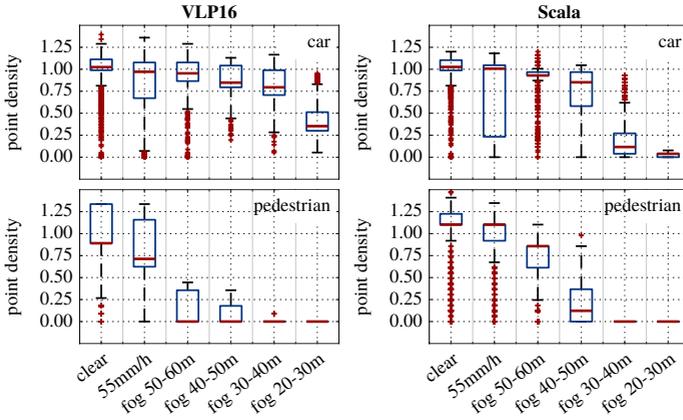


Figure 5.8: Object point density for a *car* and *pedestrian* object. The density is the number of points per object in one frame scaled by the average number of points per object in clear conditions.

5.1.3 Influence of Weather on Extracted Features

Following analysis of the influence of weather on the complete point cloud and areas that can be assigned to specific objects, this section covers the influence of weather on the extracted features, which are stated in [Section 4.1.1](#). This investigation was already published in [85]³. The analysis of the extracted features is performed using the 'chambers16' dataset which was first presented in [Section 4.2.3](#).

[Figure 5.9](#) shows selected features for more than 1, 200 frames for each weather condition of the static setup. Considering the number of points for each return $N_i(k)$, it is to be expected that the number of second and third returns will increase with the presence of fog and rain due to multiple reflections. In [Figure 5.9](#), the different weather conditions are discernible in the signal $N_{1,2}(k)$. There is a significant difference in the variance of the second echo N_2 in foggy, rainy and clear conditions. It is also interesting, that there is no significant difference in the N_3 value for dense fog and rain.

³ Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

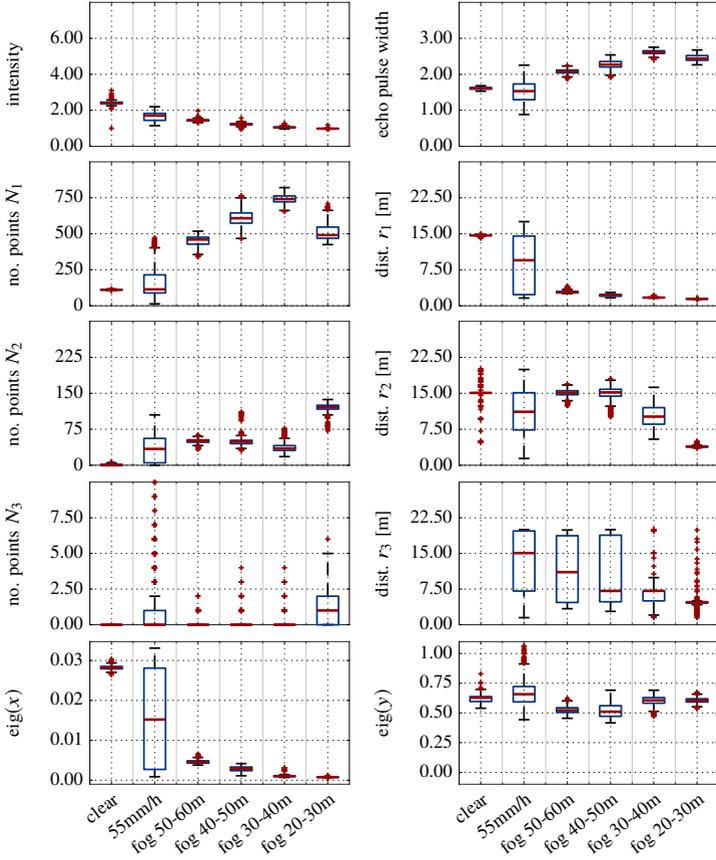


Figure 5.9: Analysis of environmental influences on LiDAR point clouds based on the *chambers16* dataset for *static scenes*. To ensure that only the influence of the weather is investigated, all the measurements represent the same *static scene*. Each column within a sub-figure denotes one distinct weather condition with at least 1, 200 frames, ordered by descending visibility. The stated mean intensity (f_{13}) is taken from the VLP16 LiDAR sensor. All other signals are based on Scala measurements, because VLP16 behavior is comparable, except for the fact that the VLP16 outputs only two echoes, which results in differences in the number of points for the second and third echoes. The intensity f_{13} , EPW f_{13} and the distances \bar{r}_t ($f_{4,5,6}$) are the mean values over all points of one frame. The number of points N_t ($f_{1,2,3}$) and mean distances \bar{r}_t are derived for each echo $t \in 1, 2, 3$ for the first, second or last return separately. The eigenvalues ($f_{13,14,16}$) were calculated from the covariance matrix of all points. The figure was published in [85].

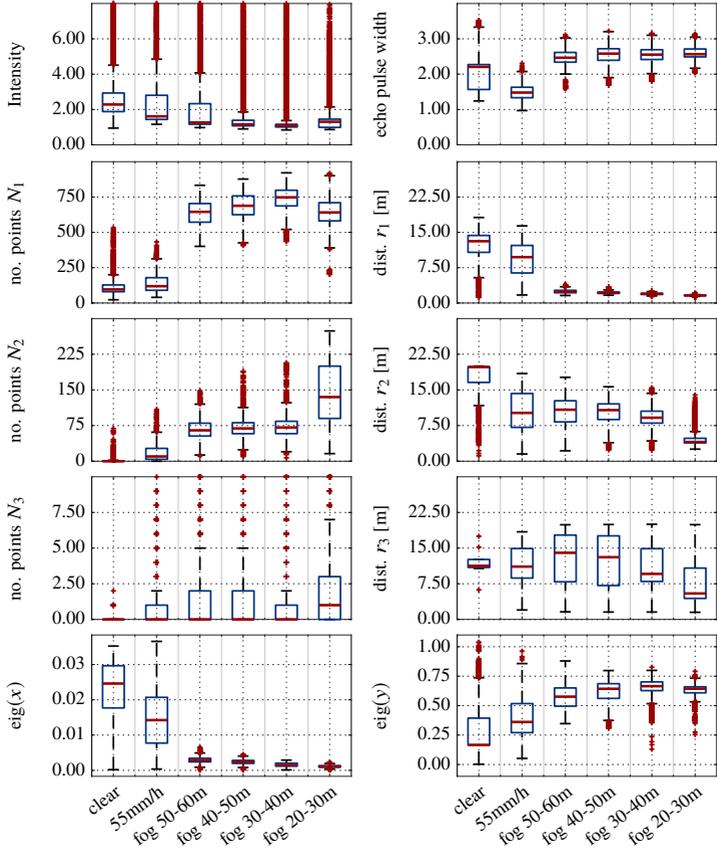


Figure 5.10: Analysis of environmental influences on LiDAR point clouds based on the *chambers16* dataset for *dynamic scenes*. The structure of the figure is identical to Figure 5.9 and was already published in [85].

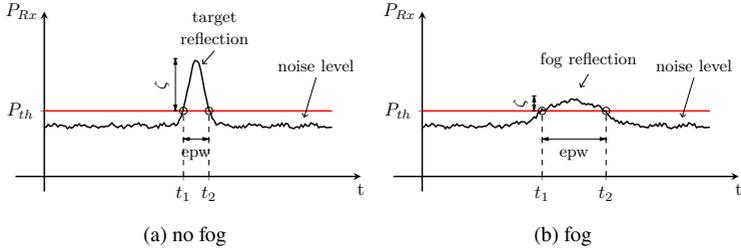


Figure 5.11: Diagram showing the influence of fog on the received back scattered light signal with a focus on the intensity and echo pulse width, which are commonly accessible in state-of-the-art LiDAR point clouds.

The mean distance \bar{r}_1 , which is calculated for each echo separately, seems to be a good measure for determining the presence of fog or rain. The described signal is illustrated in Figure 5.9 and shows a strong decrease for \bar{r}_1 in fog. At the same time, the greatest variance can be seen with the rain. The mean and variance of $r_{2,3}$ increases in rain and fog compared to clear conditions.

The paradox of increased *EPW* and reduced *intensity* by fog is discussed in [18]. Figure 5.9 confirms the claim of [18] as the *EPW* from the Scala sensor increases in foggy conditions and is approximately inversely proportional to the fog density. A diagram illustrating this effect is provided in Figure 5.11. Furthermore, reflections from rain droplets show a smaller *EPW* because water droplets in rain are less dispersed than in fog. As a result, the *EPW* is highly influenced by weather and could be used as a signal to gain information about the local environmental conditions. Regarding the intensity of the Velodyne VLP16 sensor, there is only a small decrease in intensity in dense fog and the greatest variance in rain.

Furthermore, rain and fog influence the *eigenvalues* of the covariance matrices of x and y ($f_{14,15}$). While the presence of fog and rain influences the eigenvalue $\text{cov}(x)$, no dependency on $\text{eig}(\text{cov}(y))$ can be derived. This could be based on the symmetrical structure of the setup related to the y -axis. Due to the small field of view in the z -direction, the $\text{eig}(z)$ is not evaluated. In summary, the influence of rain and fog is visible in static scenes using the LiDAR point cloud.

We will now evaluate dynamic scenarios with the same methods. A setup of dynamic scenarios is repeated for the well-controlled environmental conditions. The dynamic scenarios mimic an approaching car, crossing cyclists

and a pedestrian walking on the sidewalk. Comparing the static scenes ([Figure 5.9](#)) with the dynamic ones ([Figure 5.10](#)) the variance increases for all derived signals, while the difference of the mean decreases. In addition, the number of outliers increases significantly, especially for the intensity of the VLP16 laser scanner. This can be explained with the scenario of the approaching vehicle, since in some frames of this scenario, retro-reflective objects were in the immediate vicinity of the sensor. In conclusion, the pattern recognition task of clustering the different environmental conditions is more challenging in dynamic scenarios than in static ones.

5.1.4 Influence of Weather on Image Representation

Another way to visualize the impact of fog and rain involves the use of intensity and distance images as shown in [Figure 5.12](#). Since this representation is similar to camera images, the impact is easy for human viewers to perceive. This representation is also used as input data for CNN-based approaches, another reason why it is useful to include this data.

The upper image in [Figure 5.12](#) shows a camera reference image of the scene. The first row of point cloud images shows the intensity and distance images for reference images, split by the distance ordered first and second return. The color coding for distance is brighter for increasing distances, and darker for decreasing distances; no return is indicated with black. The intensities are color-coded in a similar manner (i.e. a brighter color denotes increasing intensity and vice versa).

In the intensity image of the first return, the zebra crossing is recognizable in the lower part of the picture; the high intensity and, thus, the yellow area in the lower left edge of the image are due to the parked vehicle and its license plate number; in the upper-right corner, the high intensity can be attributed to a traffic sign, which is not in the field of view of the camera image in [Figure 4.5a](#). However, the traffic sign is present in the upper image in [Figure 4.5c](#). The decreasing contrast with increasing fog or rain is clearly apparent, to the point where only large retroreflective elements are recognizable in the intensity image. In all instances, the information contained in the intensity of the second return is very weak.

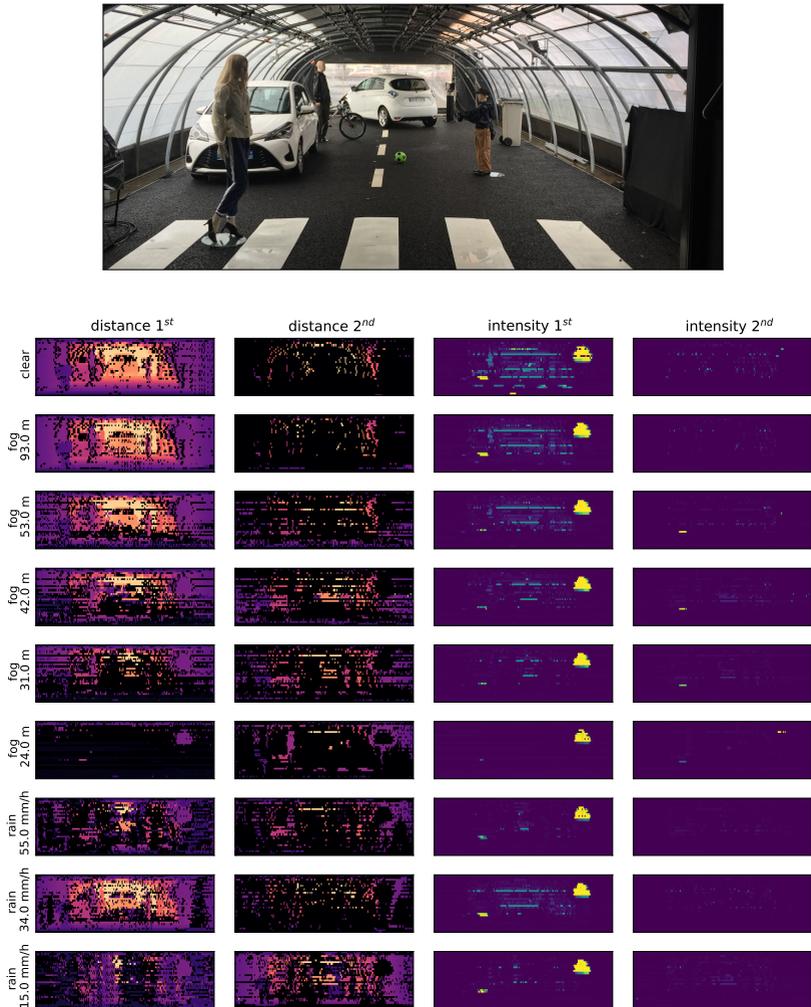


Figure 5.12: Influence of fog and rain on the LiDAR sensor's distance and intensity images. The enumeration of the returns is performed with increasing distance, whereby there is no measuring point in either returns. The sensor used here was a Velodyne VLP32C [22] which captured the static scene 1 'Pedestrian Crossing' as shown in Figure 4.5a.

Regarding the distance image under clear conditions, it can be seen that a second return occurs mostly at edges of objects, when the laser beam hits an object and divides it. In dense fog or heavier rain, the second return provides valuable information about the scenery, while the first return is caused by water droplets or disappears altogether, as in heavy dense fog (e.g. at 24 m visibility).

5.2 Weather Classification by LiDAR Sensors

This section analyzes the weather classification methods by LiDAR point cloud data only. The weather classification is based, first, on the manual extracted feature vector (see Section 4.1.1) as input data for an kNN and SVM classifier and, second, based on distance and intensity images (see Section 2.1.2) as input data for a CNN. The detailed CNN structure is presented in Section 4.1.2.

5.2.1 Weather Classification by Manually Extracted Features

For the development of a weather detection algorithm a kNN with $k = 10$ and an SVM are applied. The prediction feature vector is specified in Table 4.1. The response of the classifier is 'clear', 'fog' or 'rain'. The different visibility ranges in fog have not been taken into account for the classifier response, since the features do not differ very much in these conditions.

The different setups in the chamber are used to reduce the time correlation of the dataset. Setups A and B are therefore used for training, while setup C is used for testing. The IoU for the VLP16 is 96.40 % (kNN) and 97.14 % (SVM) and, thus, highly satisfactory. The classification result for the Scala sensor is 58.89 % for the kNN and 78.66 % for the SVM classifier and thus significantly lower than the results of the VLP16, which could be caused by the significantly smaller vertical field of view and, in turn, fewer number of points per frame. Since the number of samples per class is not evenly distributed, the accuracy is not used to evaluate the classifiers in detail as illustrated in Table 5.1. Regarding the IoU per class, the kNN approach obviously does not yield good classification results for the classes 'clear' and 'rain' for the Scala sensor. The SVM achieves slightly better results for the class 'clear' and significantly better results for the class 'rain'.

Since the weather conditions of the real-world and climate chamber data differ significantly, they are considered separately. The training and verification of the road-based data is also considered separately. Thus, four recordings under clear conditions, three with occasional rain and two with permanent rain are used for training, while the remaining recordings are used for testing purposed (one recording under clear conditions, two with rain during the day and at

Table 5.1: The overall classification testing results for climate chamber and road data. The number of samples used for testing is stated in each row per class. The **TPR**, **FPR** and the **IoU** are given as classification measures. The classes are numbered as follows: 1 clear, 2 rain and 3 fog. The classifiers with the greatest performance in terms of mean IoU are printed in bold.

place	clf	class	# samples		TPR [%]		FPR [%]		IoU [%]	
			VLP	Scala	VLP	Scala	VLP	Scala	VLP	Scala
climate chamber	kNN	1	5,558	5,643	93.91	66.47	6.09	33.53	93.85	41.37
		2	10,566	14,115	97.52	64.13	2.48	35.87	95.86	43.68
		3	92,708	101,707	99.98	94.43	0.02	5.57	99.48	91.61
	SVM	1	5,558	5,643	100.00	83.19	0.00	16.81	96.29	53.34
		2	10,566	14,115	95.86	84.92	4.14	15.08	95.78	83.70
		3	92,708	101,707	99.80	99.78	0.20	0.22	99.35	98.95
road	kNN	1	33,369	-	97.60	-	2.40	-	96.72	-
		2	4,570	-	92.45	-	7.55	-	77.04	-
	SVM	1	33,369	-	97.34	-	2.66	-	96.47	-
		2	4,570	-	92.25	-	7.75	-	75.17	-

night). The dataset was subdivided such that each dataset contains samples from every traffic scenario (empty road, traffic jam, inner city, etc.) and every weather condition and, at the same time, a subdivision of 80% to 20% is given between training and testing. This avoids time series effects. The **mIoU** achieved for the 'VLP16' is 86.88% with the kNN classifier. The IoU for the class rain is at 77.04% significantly lower than the IoU of the class 'clear' (96.72%). The decrease of the IoU for rain in real-world environments could be caused by the larger variety of rainfall rates and the lower accuracy of ground truth.

5.2.2 Weather Classification by CNN

Weather classification by **CNN** is applied to and evaluated on multiple datasets, allowing the basic capability of weather classification by sensors to be analyzed. The parameters of the proposed model are adjusted in line with the complexity of the classification task in order to optimize the classification of on the utilized data.

Due to the poor vertical resolution of the Scala and Velodyne 'VLP16' LiDAR sensors with only 4 and 16 layers respectively, the **CNN**-based classification approach is not applied to datasets recorded with these sensors.

Table 5.2: Class split for climate chamber framewise classifications. Due to the detailed ground truth information, we can split the classes rain and fog to a high level of detail.

	Weather Ground Truth	Fog Class names				Rain Class Names 3 classes
		1 class	3 classes	4 class	5 class	
Visibility	0 – 16 m	Fog1	Fog3	Fog 4	Fog 5	
	16 – 32 m		Fog2	Fog3	Fog 4	
	32 – 48 m			Fog 3	Fog 3	
	48 – 64 m		Fog1	Fog2	Fog2	
	64 – 80 m			Fog1		
	80 – 96 m			Fog1	Fog1	
	96 – 112 m					
112 – 128 m						
Rainfall Rate	15 mm/h				Rain1	
	33 mm/h				Rain2	
	55 mm/h				Rain3	

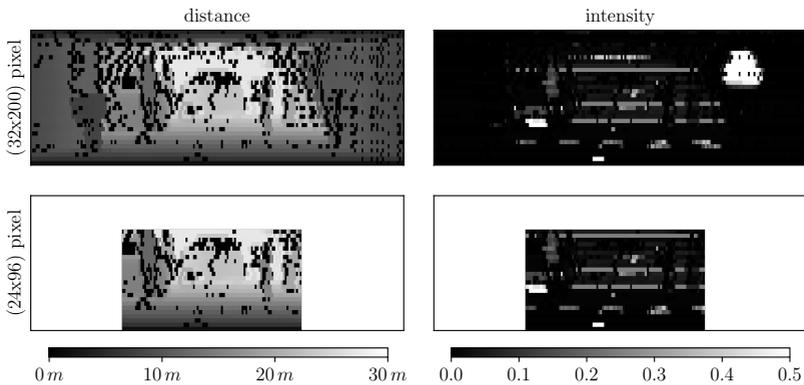


Figure 5.13: LiDAR distance and intensity image for the climate chamber scene "pedestrian crossing" shown in Figure 4.5a. The bottom images show the field of view used for CNN training in order to prevent over-fitting and to reduce the impact of the local scenery.

Climate Chamber

We first evaluate the CNN-based weather classification on the climate chamber data *chambers32* with static and dynamic scenes. For these recordings, the ground truth information is remarkably detailed, which means can split the weather classes for each weather condition even further, as shown in Table 5.2.

Table 5.3: Results of frame classification for climate chamber data with three different label mappings. The best performance in terms of IoU per column is printed in bold; the best overall is shown in blue.

Classes	Model	Clear	Fog1	Fog2	Fog3	Fog4	Fog5	Rain1	Rain2	Rain3	Mean
3	<i>WeatherClass BN</i>	94.87	93.04	–	–	–	–	97.61	–	–	95.17
	<i>WeatherClass DO</i>	90.01	88.13	–	–	–	–	97.25	–	–	91.80
	<i>RangeClass</i>	97.93	65.94	–	–	–	–	50.53	–	–	71.47
7	<i>WeatherClass BN</i>	83.80	42.26	46.59	75.55	–	–	41.45	64.82	31.20	55.10
	<i>WeatherClass DO</i>	76.07	46.54	65.12	83.33	–	–	64.43	57.51	80.80	67.68
	<i>RangeClass</i>	88.48	36.69	54.28	67.53	–	–	46.34	40.58	39.02	53.27
9	<i>WeatherClass BN</i>	58.35	16.33	26.78	38.20	62.76	77.98	24.89	50.94	36.78	43.67
	<i>WeatherClass DO</i>	54.86	18.52	25.29	41.09	68.73	81.64	55.85	44.97	60.81	50.20
	<i>RangeClass</i>	55.37	3.28	31.75	33.02	66.34	82.69	54.17	36.17	48.04	45.65

Fog and rain class splitting is performed to obtain similar data distribution in terms of sample per class. Due to the lack of data for fog densities above 80 m, however, this is not completely possible, as shown in the class distribution for the dataset in the appendix (Figure A.1a).

Section 5.1 analyzes the influence of weather on LiDAR point clouds in detail. Figure 5.2 in particular shows that the weather information contained in the point cloud is highly detailed. This information is therefore used for train a model that is capable of distinguishing among three different rain levels and up to five different fog densities. To avoid over-fitting to local dependencies, we crop the input image drastically so that the ceiling and walls of the chamber are almost invisible during CNN training as shown in Figure 5.13. In addition, the different static and dynamic scenes reflect realistic situations that correspond to road scenes and so further prevent the model from over-fitting to the climate chamber data or scenarios.

The class distribution is imbalanced, as reflected in the appendix in Figure A.1a. We therefore first evaluate the classification performance based on the IoU as presented in Section 2.5.7 as a performance metric for imbalanced datasets.

In Table 5.3 the IoU values for each class and the mIoU are given for three different CNN approaches:

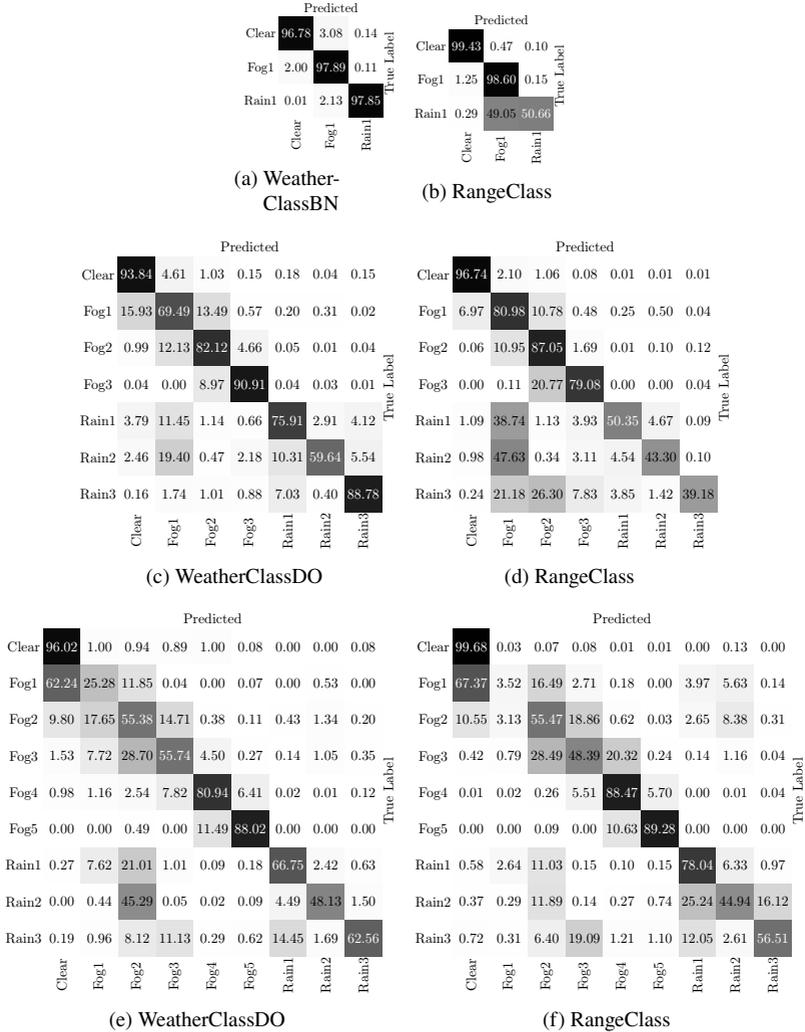


Figure 5.14: Framewise classification results for the *chamber32* dataset with three different label mappings and the CNN architectures *WeatherClass* and *RangeNet*. For the *WeatherClass* approach, the dropout variant is shown - except for the three class case - where the batch normalization is given, which corresponds to the best results of Table 5.3.

<i>WeatherClass BN</i>	Our proposed approach from Figure 4.2 with batch normalization.
<i>WeatherClass DO</i>	The same as <i>WeatherClass BN</i> , but with a dropout layer instead of batch normalization.
<i>RangeClass</i>	Backbone architecture from <i>RangeNet</i> encoder as presented in [146] , enlarged by our decoder for classification as used in <i>WeatherClass BN</i> .

For improved generalization and less over-fitting, we propose the *WeatherClass* with batch normalization or a dropout layer because the results do not clearly show a superior architecture for different evaluations.

For each approach, the **IoU** values for the three different label mappings with 3, 7 and 9 classes are given. Furthermore, the best overall values are shown in blue, while the best per column is printed in bold.

Unsurprisingly, the simpler classification tasks with a smaller number of classes and less fine class subdivision, deliver superior performance. For example, our proposed approach *WeatherClass BN* achieves an **mIoU** of up to 95 % for three classes and so yields highly convincing results. Furthermore, the **IoU** per class is very high and, for each class above 93 %. For the three-class approach, the *WeatherClass* with batch normalization outperforms the dropout variant because the normalization over an entire batch seems to yield better results than the simple dropout method.

Regarding the results with increasingly finer class subdivisions, the **mIoU** decreases as expected. To examine misclassifications in more detail, the confusion matrix is used as a metric. This allows an analysis of permuted classes to determine whether false classifications within a weather category lead to a lower **IoU**, or the weather categories are completely permuted. The first has no influence on the overall goal of recognizing the weather condition. [Figure 5.14](#) shows the confusion matrices for the three different label mappings and two different **CNN** architectures. Note that the confusion matrices in [Figure 5.14](#) have the best results in terms of **mIoU** from [Table 5.3](#).

First, the baseline experiment with three classes shows that the *WeatherNet* is in principle capable of distinguishing between the three classes 'clear', 'rain' and 'fog' ([Figure 5.15a](#)), as the **IoU** values already suggest. The *WeatherNet* outperforms the modified *RangeNet*, which incorrectly and very frequently classifies rain as fog.

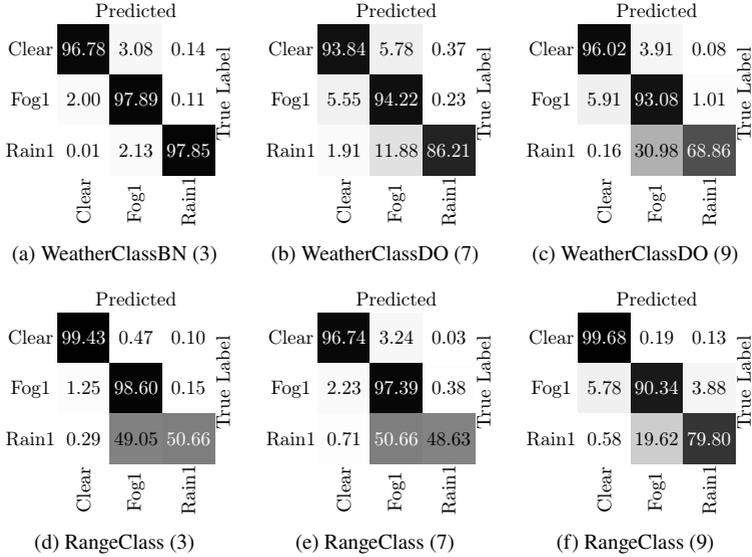


Figure 5.15: Frame-wise classification results for the *chamber32* dataset evaluated on three classes.

Furthermore, the results show that a classification of different fog densities and precipitation rates based on LiDAR data is possible. This approach is very capable of distinguishing between three different fog and rain classes, as shown in Figure 5.15b. Nevertheless, the confusion of the fog class above 80 m visibility increases with clear conditions and rain at 33 mm/h. This confusion is understandable on the basis of the input data (see Figure 5.12). For the results of *RangeNet*, the confusion is even more noticeable. However, the mix-ups are mostly in the same weather category or in the neighboring class (e.g. 'Clear' and 'Fog 1'). Since the climate chamber used is only around 25 m long, the influence of fog above visibility of 80 m can barely be seen in the range image (Figure 5.12). This explains, the confusion with the weather class 'Clear'.

To compare the performance of the different class mappings, the 7- and 9-class approaches are evaluated on the 3-class basis. Figure 5.15 shows the corresponding confusion matrices. Consequently, the *WeatherClass* approaches for 7 and 9 classes provide satisfactory results on the 3-class task. However, the results of the network trained and evaluated on 3-classes reveal the best results. For the 9-class approach in particular the confusion between rain and

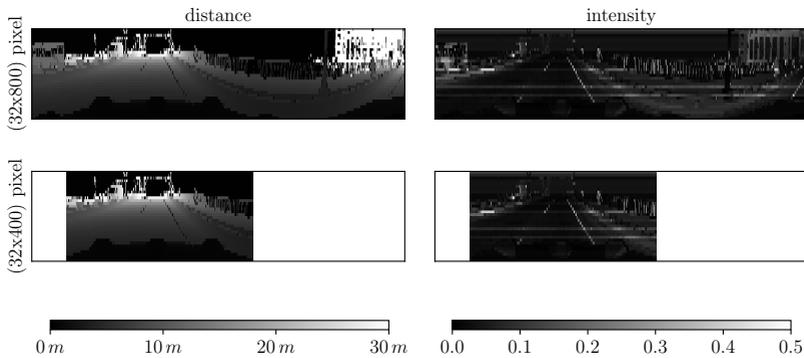


Figure 5.16: LiDAR distance image for the road data 'NuScenes'. The bottom images show the field of view used for CNN training in order to focus on the forward facing view.

fog increases. In contrast, the *RangeNet* trained and 9-classes and evaluated for 3-classes provides the best results for this architecture and less confusion among rain, fog and clear conditions. In conclusion, weather classification based on LiDAR point cloud data with CNN approaches yields highly satisfactory results and even predicts precise increments based on meteorological metrics within identical weather conditions. Down-sampling in the *RangeNet* architecture aims to understanding the scene globally but is not suited to the task of weather classification for the the climate chamber data, where weather effects partially appear inhomogeneously, such as fog or the artificial rain in the climate chamber used. In contrast, our proposed *WeatherNet* - without any downsampling layers - is capable of retrieving the weather information from any fine structure of the input data and thus predict weather conditions in great detail, albeit with some confusion within the same weather condition. The results also reveal that the batch normalization aiming for better generalization and less over-fitting does not provide the best results for the 7- and 9-class experiment. This might be due to the low sample counts per class and the significant alterations of the input data in case of e.g. very dense fog, whereby the normalization can vary significantly with each batch. In such cases, the dropout layer provides better results (7- and 9-class experiment) for the experiments on the chamber data.

Table 5.4: Results regarding frame classification for the NuScenes dataset. The best performance in terms of IoU per column is shown blue; the second best is printed in bold.

Model Name	Layer Depth	Clear	Rain	Mean
<i>Baseline</i>	2,4,8	86.14	87.83	86.99
<i>NuScenes-WeatherNetDO</i>	2,4,8,16	90.54	91.34	90.94
<i>NuScenes-WeatherNetBN</i>	2,4,8,16	96.68	96.77	96.72
<i>NuScenes-WeatherNetDO</i>	4,8,16,32	95.61	95.74	95.68
<i>NuScenes-WeatherNetBN</i>	4,8,16,32	96.78	96.82	96.80
<i>NuScenes-RangeNet21</i>	2,4,8,16	98.72	98.72	98.72
<i>NuScenes-RangeNet21</i>	2,4,8	97.51	97.52	97.52
<i>NuScenes-RangeNet21</i>	4,64	99.26	99.26	99.26

Nuscenes Data Set

To analyze the proposed CNN-based weather classification approach on road data, we use the *NuScenes* dataset. Since the influence of the weather is independent of the field of view, we crop the input data to a forward-facing viewing region, as shown in [Figure 5.16](#). The CNN architectures used are again *WeatherClass BN*, *WeatherClass DO* and *RangeClass*; these were introduced in previous section. Given that the classification task is less difficult in terms of the number of classes, the depth of each filter and the number of layers are reduced as long as the validation results are not affected. Furthermore, the resulting number of filters per layer is optimized, which means that the overall validation results remain without degradation. In principle, the task of detecting whether or not it is raining is a much simpler 2-class problem than the more complex task of detecting different weather conditions in detail. This fact is reflected in the simplified network architecture without any compromise in performance.

The results can be found in [Table 5.4](#), whereby the baseline represents a network architecture with single convolution layers. The best result in terms of [mIoU](#) is obtained with the *RangeClass* approach with two layers to a depth of 4 and 64. The corresponding [IoU](#) results for each class are very high (99.26 % for both classes). For the *WeatherClass BN* architecture, the best results are 96.78 % ('Clear'), 96.82 % ('Rain') and 96.80 % for the [mIoU](#), by applying four layers to depths of 4, 8, 16 and 32. By significantly reducing the number

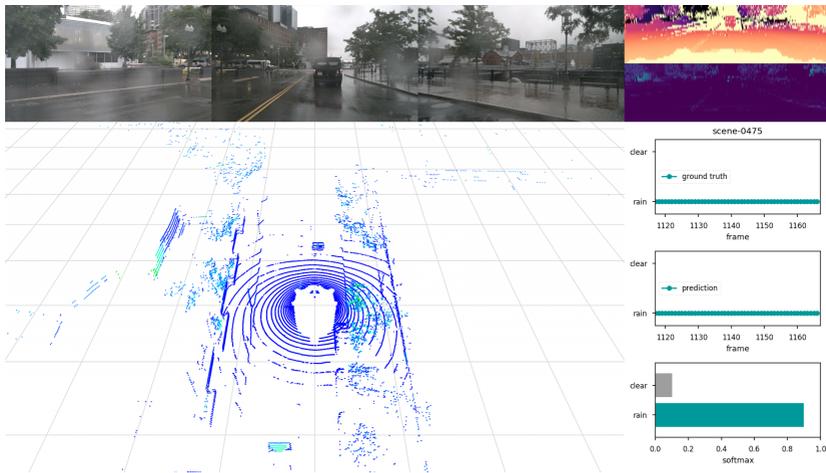


Figure 5.17: Experimental results on Nuscenes dataset. The top row shows the forward facing camera images, whereas the top right images are representing the LiDAR distance and intensity image. In the center of the figure the point cloud is shown, color coded by intensity. The plots on the right side shows the history of the ground truth, prediction and the corresponding softmax.

of layers and filters, the resulting size of the CNN is reduced and so requires much less computational power. This is also demonstrated by the baseline experiment.

Figure 5.17 shows the classification result on a representative snapshot of 'scene-0475' with the camera reference images (top left), point cloud (center), distance and intensity images (top right). The results of the CNN can be seen on the right side, along with the history of the ground truth and prediction. Note only the distance and intensity image is fed into the CNN for classifying the weather condition.

The result with the *NuScenes* dataset reveals that no one CNN architecture is superior with all datasets. Since the classification task and data structure itself differ very significantly in both experiments, the CNN architecture requires some optimization in order to achieve optimal results. In the case of the *NuScenes* experiment, the alteration of the input data, which were corrupted by rain, is homogeneous over the entire field of view, at least for the parts that were fed into the CNN (see Figure 5.17 and Figure 2.4). For the chamber data - and especially with fog - this is not the case (see Figure 5.12). As a result,

the *RangeNet* architecture yields better results with the *NuScenes* dataset by down-sampling the input image size and retrieving global scene understanding. For the climate chamber data, deriving a global understanding of the scene is impractical due to the inhomogeneous nature of the weather effect in the input data space. The network architecture without down-sampling of the input data space (*WeatherClass*) is capable of detecting these inhomogeneous effects and so yields better results.

In summary, for weather detection with homogeneous influences on the 2D input data (e.g. whether or not it is raining) an architecture with down-sampling - such as *RangeNet* - is preferable. That said, for weather with inhomogeneous effects such as fog or spray, architectures without down-sampling - such as *WeatherClass* - are more effective.

5.3 Pointwise Weather Segmentation

This section was previously published in [86] and [87] and contains information about the experiments for pointwise weather segmentation for sparse LiDAR point clouds⁴. In contrast to Section 5.2, where a class label is obtained for a single frame, this section determines a class label for each point with respect to the weather condition. The classification of a single point - i.e. a single pixel - is known as segmentation in computer vision, which is why we introduced the term *weather segmentation* to distinguish from a frame-by-frame classification approach.

5.3.1 Experiments

As described in Section 4.1.3 and 4.2.3, we obtained a large-scale dataset recorded on public roads and in a dedicated climate chamber with different types of point-wise annotations. In this section we will describe a number of approaches for training the proposed *WeatherNet* in order to maximize the performance and analyze the benefit of weather augmentation, especially for the generalization to natural rainfall recorded on roads. We apply the IoU metric for performance evaluation, according to the Cityscapes Benchmark Suite [83, 145]. An overview of all experiments and their results is given in Table 5.5. To evaluate the influence of weather augmentation in detail, we trained the network on three different data subsets with and without augmentation (defined as experiments 1, 2 and 3):

1. **Chamber:** only chamber data as baseline experiment.
2. **Chamber and road:** Climate chamber dataset and a subset of road data without any augmentation or adverse weather on roads.
3. **Chamber and road with augmentation:** Climate chamber data set and class-balanced road dataset without adverse weather, but with augmentation.

⁴ Editorial modifications have been made to improve readability and ensure consistency of nomenclature.

Table 5.5: Results on the test data set. The best performance in terms of IoU per column is printed in bold, the overall greatest in blue.

Approach	Experiment 1			IoU in % / Number of Samples Experiment 2			Experiment 3			Runtime ³⁾ in		Parameter ⁴⁾ in Mio		
	Clear	Fog	Rain	Clear	Fog	Rain	Clear	Fog	Rain	Mean	Mean			
Samples Chamber ¹⁾	14,386	29,777	28,637	72,800	14,386	29,777	28,637	72,800	14,386	29,777	28,637	72,800	-	-
Samples Road ^{1),2)}	-	-	-	-	31,078	-	-	-	34,381	34,381	34,381	103,143	-	-
<i>DROR</i> [123]	88.13	6.94	7.37	34.15	88.13	6.94	7.37	34.15	88.13	6.94	7.37	34.15	100.00	4e-6
<i>RangeNetv3</i> [146]	74.73	77.32	91.22	81.09	87.75	86.46	94.23	89.48	86.50	87.19	94.36	89.35	51.90	66.17
<i>RangeNet2</i> [146]	71.53	71.40	86.13	76.35	86.71	80.90	87.01	84.87	85.10	79.94	85.35	83.46	33.83	38.50
<i>LiLaNet</i> [145]	82.72	79.57	88.16	83.48	91.60	84.96	88.62	88.39	93.85	88.74	90.82	91.14	91.93	7.84
<i>WeatherNet</i> [Ours]	91.65	86.40	89.29	89.11	90.89	85.15	88.84	88.29	93.35	88.81	90.92	91.03	34.45	1.53

¹⁾ the column *Mean* states the total number the samples ²⁾ fog and rain are augmented samples ³⁾ on GeForce GTX 1080 Ti ⁴⁾ number of trainable parameters

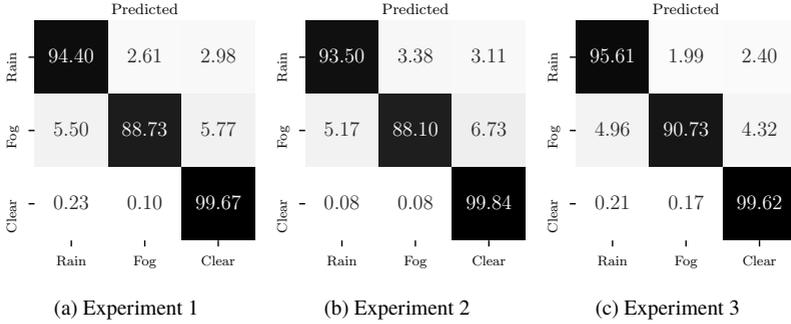


Figure 5.18: Confusion matrix for *WeatherNet* pointwise segmentation results. For each experiment the confusion matrix is shown separately.

5.3.2 Results with Static Chamber Data

Note that all evaluations are based on the test dataset from experiment 2 'Chamber and Road', which contains auto-labeled annotations and road data without fog, rain or augmentation. Table 5.5 shows that the performance is significantly increased by using road data and the proposed weather augmentation. In addition to validating the classes 'fog' and 'rain' only on chamber data, the use of road data and augmentation leads to an increase in the overall performance and per-class IoU. This indicates that the network is capable of identifying weather influences in both domains and gains a general understanding of the scene.

The results of the baseline *DROR* filter in Table 5.5 indicate that sparsity might not be a valid criterion by which to filter scatter caused by fog or drizzle as soon as distribution density of water drops increases. The proposed *CNN* approach outperforms *DROR* by an order of magnitude. In conclusion, the evaluation demonstrates that filter approaches are prone to failure in the near and far range when only the spatial neighborhood is used. The parameters for the *DROR* are taken from [123] - with the exception of the horizontal sensor resolution, which is adapted in line with 'VLP32C'.

We then compare our approach to the state-of-the-art semantic segmentation models *RangeNet21*, *RangeNet53* [146] and *LiLaNet* [137], which provide comparable results. We can therefore prove that the basic idea of *CNN*-based weather segmentation and de-noising is valuable and is superior to geometrically based approaches. In addition, the proposed optimized *WeatherNet*

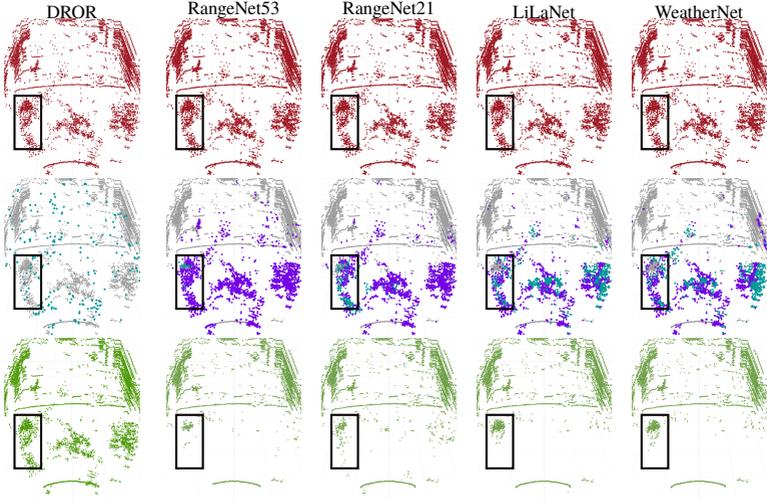
mostly outperforms the other *CNNs* - especially in the final experiment 3 - and has a significantly lower number of trainable parameters and inference time. The network can therefore be applied as a pre-processing step.

The confusion matrices in [Figure 5.18](#) show that the classes 'rain' and 'fog' are most likely to be mixed up. This is hardly surprising since fog and rain ultimately consist of water droplets and differ only in terms of the distribution, density and size of the water droplets. Moreover, [LiDAR](#) sensors are not designed to detect this difference. For point cloud filtering, these mix-ups are not important; any such confusion is a problem only with regard the classification of distinct weather conditions. Furthermore augmentation results in a significant decrease in confusion between rain and fog.

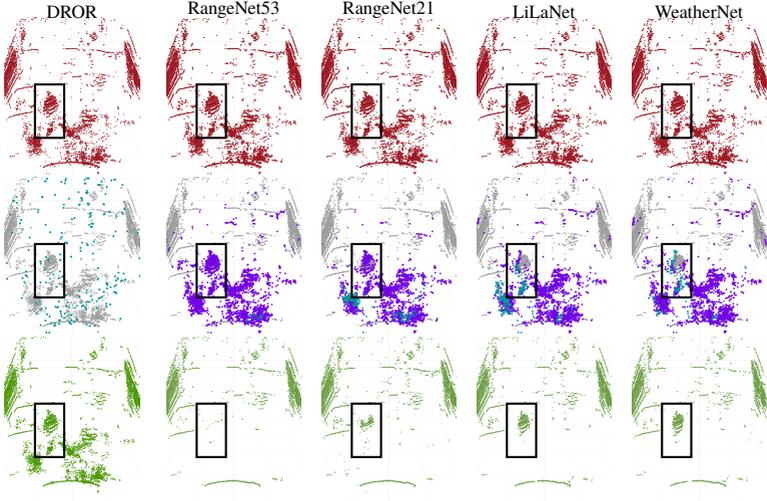
5.3.3 Results with Dynamic Chamber Data

This section presents qualitative results on challenging dynamic scenes, although no ground truth data is available because the very same dynamic scenes cannot be recorded under two different weather conditions. Our proposed auto-labeling procedure therefore cannot be applied. Nevertheless, the three representative results shown in [Figure 5.19](#), [Figure 5.20](#) and [Figure 5.21](#) show that our approach is capable of handling dynamic scenes in various domains and yields exceptionally good filter results. The results therefore demonstrate that the network can generalize and is capable of distinguishing [VRU](#) from the weather influence of fog or rain even in the close range of the sensor. Consequently, in the examples presented, the [VRU](#) remains after filtering, while the points due to fog and rain are filtered.

The de-noised point cloud reveals a pedestrian and a cyclist (highlighted by black boxes) directly in front of the ego-vehicle, who almost disappear in the scatter points of the haze ([Figure 5.19](#), [Figure 5.20](#)). Although the evaluated performance of *RangeNet53* and *WeatherNet* are comparable ([Table 5.5](#)), the qualitative results show that *RangeNet53* does not preserve fine structures or the edges of small objects ([Figure 5.20b](#), [Figure 5.19a](#) or [Figure 5.19b](#)), since most parts of the cyclist and pedestrian are filtered. *WeatherNet*, however, is capable of distinguishing between the pedestrian/cyclist and scatter points and so preserves the fine structures due to a network architecture without down-sampling layer.



(a) The highlighted cyclist is riding in fog at 30 m visibility.



■ Input ■ Valid ■ Fog ■ Rain ■ De-Noised

(b) The highlighted pedestrian is walking in fog at 20 m visibility.

Figure 5.19: De-noising results shown on two snapshots. Note that the **VRU** remains after filtering by *LiLaNet* or *WeatherNet* while the fog clutter is discarded.

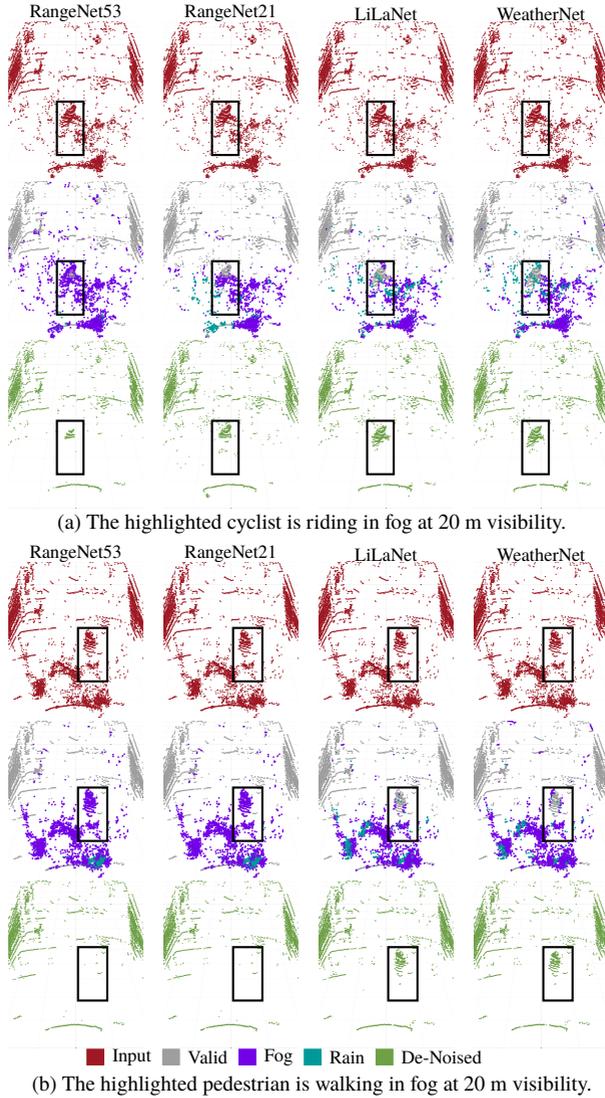


Figure 5.20: De-noising results shown on two snapshots. Note that the **VRU** remains after filtering by *LiLaNet* or *WeatherNet* while the fog clutter is discarded.

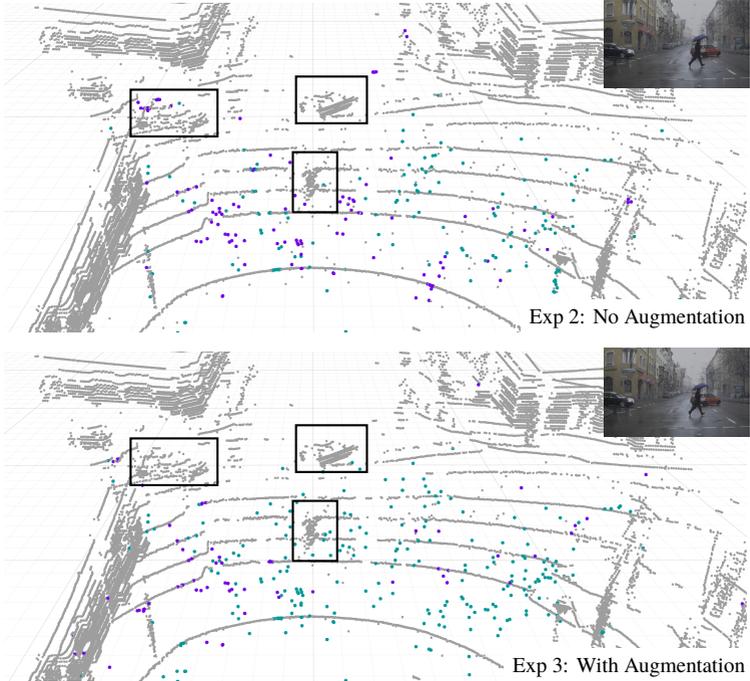


Figure 5.21: *WeatherNet* segmentation results for road data recorded under light rainfall. The result shows, that the approach is able to generalize the noise pattern and predicts well results for recordings during natural rainfall. Note that the training data set neither contains natural rain nor fog scenes on roads. The color coding is similar to [Figure 5.20](#) and the objects shown in the camera image are highlighted with a black box. The training with augmentation (right) leads to a better segmentation result in terms of number of detected raindrops and less false negatives for object detection (e.g. left car).

A filter algorithm based on the spatial distribution of the point cloud (see [Figure 5.19](#)) is not capable of filtering the noise in this scenario, since the fog points are as densely distributed as those of real objects. Nevertheless, the cyclist can be recognized slightly better, thanks to the ability to filter single scattering points. Note that the [DROR](#) also filters various single points at greater distances and that are not caused by the weather due to the geometric approach of filtering single points without neighboring points in the vicinity.

Another benefit of our *CNN* approach is its capability of detecting weather conditions by means of [LiDAR](#) point clouds. As shown in [Figure 5.2](#), the number of scatter points caused by fog is correlated with the meteorological visibility, which means that visibility can be estimated on the basis of the result of our weather segmentation. Moreover, the level of degradation of the [LiDAR](#) sensor could be estimated by taking into account the ratio of scatter to valid points.

5.3.4 Results with Dynamic Road Data

Additionally, the proposed approach is capable of processing a point cloud corrupted by natural rainfall recorded on roads. [Figure 5.21](#) shows a key frame where a pedestrian is crossing the street and several cars are passing by. Despite the fact that the algorithm was trained only with data acquired in the climate chamber and with augmented data from real-life traffic scenarios, it performs very well in road traffic under real-life conditions with light rain and so proves the generalization to another domain, as can be seen in [Figure 5.21](#). In addition, augmentation improves the result in that significantly fewer individual points of the highlighted objects are falsely recognized as rain or fog. Furthermore, more points are correctly classified as rain ([cyan](#) points) and not mixed up with fog ([purple](#) points).

5.3.5 Visibility Estimation

As shown in [Figure 5.2](#), the point cloud combined with the ground-truth annotations for each single point reflects in great detail the fog density with respect to visibility. The pointwise prediction of the class labels 'clear' or 'fog' therefore enables visibility to be estimated on the basis of the point cloud input

data. To yield an end-to-end visibility estimation based on the point cloud data, however, we have to adapt the aforementioned *WeatherNet*. We will therefore introduce a regression head attached to the proposed CNN approach, which is capable of predicting an MOR between 5 m and 100 m visibility based on the point cloud data obtained in the climate chamber.

Due to the timescale of this work, the purpose of the approach is purely to demonstrate the possibility of estimating the MOR by applying the proposed architecture. It is not the goal to present a complete optimization of different CNN architectures, review state-of-the-art regression models for the purpose of visibility estimation or perform a detailed evaluation of accuracy compared with other state-of-the-art methods. The results are simply discussed and evaluated on the basis of Figure 5.22 and demonstrate extent to which it is possible to estimate the MOR based on LiDAR point clouds. As Figure 5.22 shows, the results are summarized in one plot, whereas the ground truth and prediction are presented. Each prediction result is grouped based on ground truth visibility in 1 m increments and plotted with the mean and standard deviation in the red curve. The corresponding number of samples is shown in gray in the histogram.

Up to a visibility of 60 m, the visibility estimation is quite accurate, but slightly overestimated. For greater visibilities, the estimation is far too inaccurate and deviates from the ground truth. This can be explained, on the one hand, by the significantly fewer samples for these visibility ranges; on the other hand, in the climate chamber with a length of only about 25 m, the influence of fog with a visibility greater than 60 m is no longer recognizable in the point cloud. This can also be seen in Figure 5.3 and Figure 5.12.

In conclusion, the proposed approach enables not only segmentation but also visibility estimation based on point clouds captured by LiDAR sensors, as long as sufficient data is available for the corresponding use case.

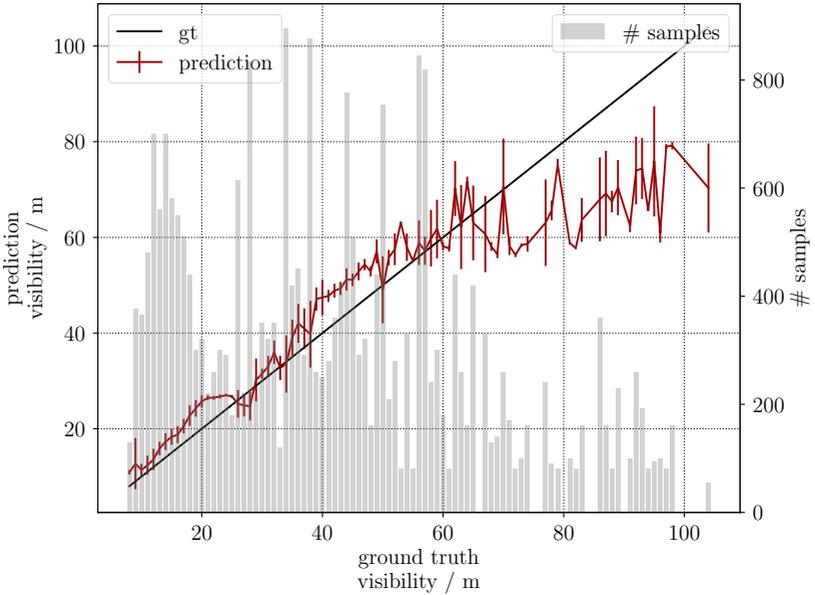


Figure 5.22: Estimation of the MOR based on LiDAR point-cloud data. The x -axis shows the ground truth visibility obtained by the climate chamber reference system [62]. The y -axis shows the predicted visibility based on the point cloud data of a Velodyne VLP32c. The ground truth (gt) data is plotted with a black line, the prediction in red. The corresponding number of samples per visibility bin is shown in light grey in the bar chart.

6 Conclusions and Outlook

6.1 Summary and Conclusion

Autonomous driving requires robust and comprehensive environment perception capability under a variety of conceivable environmental conditions like heavy rain and dense fog. In addition, a semantic understanding of the scene is required for providing valuable information for subsequent signal processing such as sensor fusion. We have shown that adverse weather has a significant impact on LiDAR point cloud quality. While this might seem obvious, it is not reflected in the public datasets currently available. We therefore recorded the first LiDAR point cloud data set with an emphasis on adverse weather, enabling in-depth analysis¹. The dataset acquired under controlled conditions contains four remarkably realistic *static* scenes in 73,000 frames and *dynamic* scenes with real pedestrians, cyclists and cars with six different complex traffic trajectories in 35,000 frames. The highly accurate ground truth weather information was provided by the chamber's closed-loop-controlled visibility from 5 m to 100 m and the precipitation intensity from 15 mm/h to 55 mm/h. We also complemented the recordings with per-point weather labels.

This comprehensive dataset enables further optimization, development and calibration of augmentation algorithms for adding realistic weather impacts on LiDAR point clouds. We optimized a state-of-the-art algorithm to create more realistic fog and introduced the first-ever - to the best of our knowledge - rain augmentation for LiDAR sensors.

We leveraged available, newly recorded and augmented data to perform an in-depth analysis of the influence of adverse weather on LiDAR sensors for robotics and automotive applications. In conclusion, we quantified - for the first

¹ The datasets were collected within the scope of this work or with the help of the publications of [155] and [157].

time to the best of our knowledge - sensor degradation based on representative objects. As an example, the perception of a pedestrian at a distance of 18 m in front of the sensor gradually degraded to approximately 50 m meteorological visibility, after which perception capability deteriorates dramatically. It is therefore apparent that **LiDAR** sensors do not fulfill the desired functionality of a robust sensor that can operate flawlessly in any weather. To assess the current perception performance, therefore, each **LiDAR** sensor needs to take into account the current environmental conditions.

We developed a novel approach in order to classify the weather status based solely on the **LiDAR** point cloud. We demonstrated a mean **IoU** of 95.17% and 99.25% for controlled and uncontrolled environments respectively.

While it is extremely important to accurately classify weather and current performance, ensuring safe operation is even more crucial. We investigated various approaches for de-noising point clouds to enable robust perception even in dense fog or heavy rain. State-of-the-art geometric approaches failed to differentiate among points caused by solid objects or weather. Our CNN-based implementation is capable of distinguishing between solid objects and weather impacts. It can even separate a moving pedestrian from a waft of fog within a single frame. We also demonstrated that the proposed **CNN** approach is capable of estimating the **MOR** based solely on the **LiDAR** point cloud data.

In addition, the dataset reveals in detail in which situations and weather conditions state-of-the-art **LiDAR** sensors fail to provide realistic representations of the environment. Detailed information on the meteorological conditions and the various realistic scenarios enable comprehensive evaluations and detailed statements regarding in the presence of which objects and under which weather conditions the sensors fail to correctly perceive the environment.

6.2 Outlook

Recognizing and understanding adverse weather impacts on sensor performance is crucial to the ability to operate autonomous systems in these weather conditions. Combining this information in the subsequent fusion of several

different sensor types enables the system to react appropriately to adverse weather impacts in the planning and control steps.

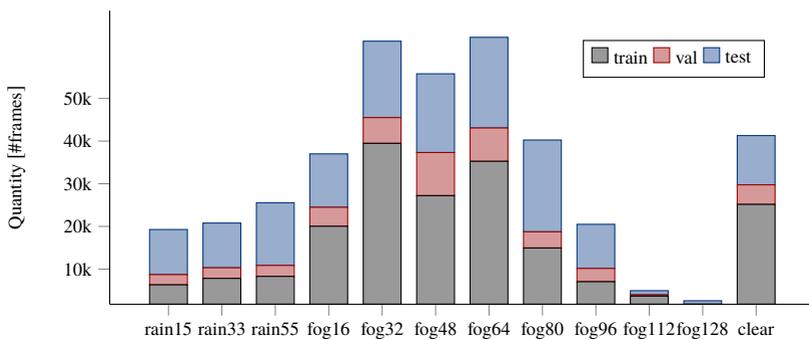
Further extensions of our work can be achieved by applying an accumulation over time for the classification methods. This can also result in a finer division of classes, which could potentially enable the intensity of rain or fog to be precisely determined. In addition, the approach for sparse point cloud denoising could be enhanced by the application of network structures which take into account temporal dependencies like [RNN](#).

In the context of road-based vehicles, the results of this work could be used to warn drivers of adverse weather and prompt them to reduce their speed accordingly. This alert could be issued to not only the vehicle that detected the adverse weather but also to the vehicles following it, with the data being transferred by vehicle-to-vehicle communication. For example, partially autonomous vehicles following the vehicle that detected the adverse weather could send a request to the driver to take over the task of driving and slow down accordingly.

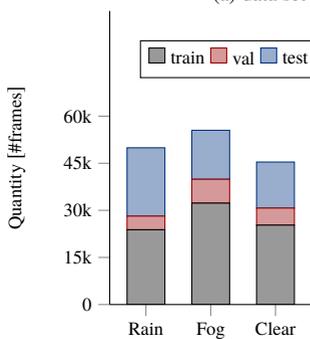
Moreover, if multiple vehicles in different locations are capable of providing valuable weather information, the resulting data could potentially prove extremely valuable for future weather analysis and forecasting systems. Due to the large number of these sensors and the regional dispersion, forecasts about the temporal and local occurrence of local rain showers, for example, can be made with much greater precision. In summary, temporal and local weather data is of enormous value even for applications beyond autonomous driving and could potentially be leveraged for numerous applications without the need for any additional hardware.

Regarding sensor development, the results of this work could potentially be used to improve the robustness of future [LiDAR](#) sensors with respect to weather effects and to extend data representations to include even more valuable information about the backscattered light. This information can be leveraged by algorithms in order to determine the weather condition or the nature of the object from which the light was reflected.

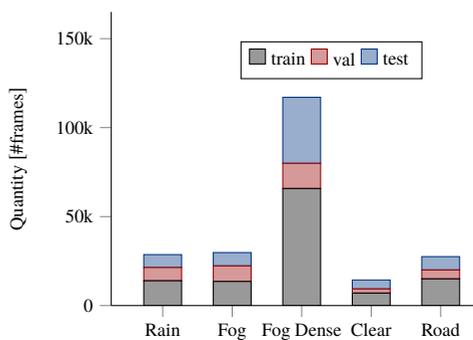
A Appendix



(a) data set *chamber32*, fine weather conditions



(b) *chamber32* dynamic



(c) *chamber32* static

Figure A.1: Overview about the class distributions for the *chamber32* data sets with dynamic and static scenes.

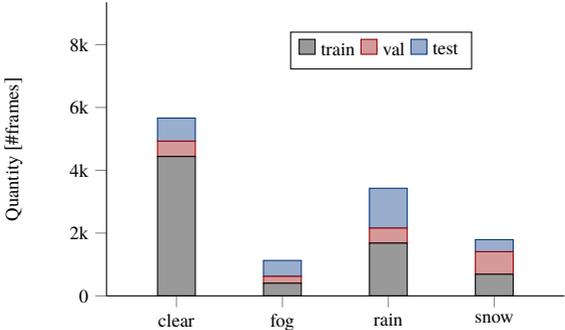


Figure A.2: Overview about the class distributions for data set *dense64*

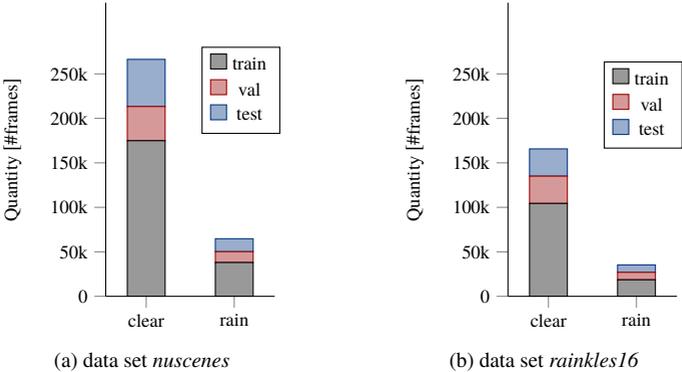


Figure A.3: Overview about the class distributions for data set used for frame-wise classifications.

Bibliography

- [1] W. H. O. (WHO), *GLOBAL STATUS REPORT ON ROAD SAFETY*. 2018. (Cited on page 1)
- [2] European Aviation Safety Agency, “Airbus A380 TYPE-CERTIFICATE DATA SHEET,” tech. rep., 2015. (Cited on page 1)
- [3] S. Bundesamt, “Verkehrsunfaelle Zeitreihen.” <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-zeitreihen-pdf-5462403.pdf>, 2019. (Cited on page 1)
- [4] Statistisches Bundesamt (Destatis), “Größte Schwierigkeiten deutscher Autofahrer im Straßenverkehr 2015.” 2020-04-14, 2019. (Cited on pages 1, 2, and 5)
- [5] SAE On-Road Automated Vehicle Standards Committee, “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems.” http://www.sae.org/misc/pdfs/automated_driving.pdf, 2014. (Cited on page 2)
- [6] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, and P. Mahoney, “Stanley: The robot that won the DARPA Grand Challenge.,” *J. Field Robotics*, vol. 23, pp. 661–692, 2006. (Cited on page 3)
- [7] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Hertrich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoeppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, “Making Bertha Drive-An Autonomous Journey on a Historic

- Route,” *IEEE intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014. (Cited on page 3)
- [8] Aeva Inc., “Aeva.” <https://www.aeva.ai/>. (Cited on pages 4 and 10)
- [9] Baraja Pty Ltd., “No Title.” <https://www.baraja.com>, 2021. (Cited on pages 4 and 10)
- [10] Blickfeld GmbH, “Blickfeld - Scan your world | Solid-State LiDAR Sensorik.” <https://www.blickfeld.com/de/>. (Cited on pages 4 and 10)
- [11] Hesai Technology, “HESAI.” <https://www.hesaitech.com/en>. (Cited on pages 4 and 10)
- [12] Innoviz Technologies Ltd, “Innoviz | Enabling the Autonomous Vehicle Revolution.” <https://innoviz.tech/>. (Cited on pages 4 and 10)
- [13] VALEO, “Valeo Scala: a laser scanner for highly automated driving.” <https://www.valeo.com/en/valeo-scala/>. (Cited on pages 4 and 10)
- [14] Velodyne Lidar Inc., “Smart Powerful Lidar Solutions | Velodyne Lidar.” <https://velodynelidar.com/>. (Cited on pages 4 and 10)
- [15] Waymo LLC, “Laser Bear.” <https://waymo.com/lidar/>, 2021. (Cited on pages 4 and 10)
- [16] N. Hautière, J.-p. P. Tarel, J. Lavenant, D. Aubert, N. Hauti, and J.-p. P. Tarel, “Automatic fog detection and estimation of visibility distance through use of an onboard camera,” *Machine Vision and Applications*, vol. 17, pp. 8–20, apr 2006. (Cited on pages 4, 21, and 24)
- [17] U. Wandinger, *Introduction to Lidar*, pp. 1–18. New York, NY: Springer New York, 2005. (Cited on page 10)
- [18] H. Gotzig and G. Geduld, “Automotive LIDAR,” in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort* (H. Winner, S. Hakuli, F. Lotz, and C. Singer, eds.), pp. 405–430, Springer International Publishing, 2015. (Cited on pages 10 and 99)
- [19] S. Bogatscher, *Time-of-Flight Laserkamera mit großer Empfangsapertur*. PhD thesis, Karlsruhe, Karlsruher Institut für Technologie (KIT), Diss., 2014, 2015. (Cited on pages 10 and 11)

-
- [20] Velodyne Lidar Inc., “Puck: VERSATILE REAL-TIME LIDAR SENSOR PUCK.” https://www.velodynelidar.com/download_s.html, 2019. (Cited on pages 12 and 72)
- [21] Velodyne Lidar Inc., “HDL-32E: HIGH RESOLUTION REAL-TIME 3D LIDAR SENSOR HDL-32E.” <https://www.velodynelidar.com/downloads.html>. (Cited on pages 12, 14, 72, and 76)
- [22] Velodyne Lidar Inc., “ULTRA Puck: HIGH RESOLUTION REAL-TIME LIDAR FOR AUTONOMOUS SYSTEMS Ultra.” <https://www.velodynelidar.com/downloads.html>, 2019. (Cited on pages 12, 72, 89, and 101)
- [23] Velodyne Lidar Inc., “HDL-64E~S3: HIGH DEFINITION REAL-TIME 3D LIDAR HDL-64E,” (Cited on pages 12 and 72)
- [24] Ibeo Automotive Systems GmbH, “ibeo . HAD Feature Fusion.” <https://www.ibeo-as.com/had/ibeo-feature-fusion/>, 2019. (Cited on pages 12, 72, and 89)
- [25] Velodyne Lidar Inc, “USER’S MANUAL AND PROGRAMMING GUIDE HDL-64E S3.” <https://velodynelidar.com/hdl-64e.html>, 2019. (Cited on pages 12, 13, and 72)
- [26] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, and Q. Xu, “nuScenes : A multimodal dataset for autonomous driving,” *arXiv preprint arXiv:1903.11027*, 2019. (Cited on pages 14, 58, 60, 71, 72, 76, 77, and 78)
- [27] M. Schillinger, *Analysis and Optimization of Fully Convolutional Neural Networks fir LiDAR-based Semantic Labeling*. masterthesis, Hochschule für Technik Stuttgart, 2018. (Cited on page 15)
- [28] R. Nebuloni and C. Capsoni, “Effects of Adverse Weather on Free Space Optics,” in *Optical Wireless Communications*, pp. 47–68, Springer, 2016. (Cited on pages 17, 18, 19, 20, 21, 22, 23, 25, and 27)
- [29] H. R. Pruppacher and J. D. Klett, “Microstructure of atmospheric clouds and precipitation,” in *Microphysics of Clouds and Precipitation*, pp. 10–73, Springer, 2010. (Cited on pages 17, 19, 20, 21, 22, 23, 26, and 27)
- [30] H. C. Hulst and H. C. van de Hulst, *Light scattering by small particles*. Courier Corporation, 1981. (Cited on pages 17 and 18)
- [31] Z. Ye, X. Jiang, and Z. Wang, “Measurements of particle size distribution based on Mie scattering theory and Markov chain inversion algorithm,” *Journal of Software*, 2012. (Cited on page 18)

- [32] G. M. Hale and M. R. Querry, "Optical constants of water in the 200-nm to 200- μ m wavelength region," *Applied optics*, vol. 12, no. 3, pp. 555–563, 1973. (Cited on page 19)
- [33] B. J. Sumlin, W. R. Heinson, and R. K. Chakrabarty, "Retrieving the aerosol complex refractive index using PyMieScatt: A Mie computational package with visualization capabilities," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 205, pp. 127–134, jan 2018. (Cited on page 19)
- [34] D. Grieser, "Divergente und asymptotische Reihen: Einführung Seminar, WS 2013/14," 2013. (Cited on page 20)
- [35] W. E. K. Middleton and V. Twersky, "Vision Through the Atmosphere," *Physics Today*, 1954. (Cited on pages 21 and 24)
- [36] W. R. Cotton and S. Yuter, "Principles of cloud and precipitation formation," in *Aerosol Pollution Impact on Precipitation*, pp. 13–43, Springer, 2009. (Cited on pages 21 and 22)
- [37] American Meteorological Society, "Fog - AMS Glossary." <http://glossary.ametsoc.org/wiki/Fog>, 2012. (Cited on pages 21, 22, and 74)
- [38] Tverskoui, *Physics of the atmosphere, a course in meteorology*. Israel program for scientific translations, 1965. (Cited on page 22)
- [39] C. D. Ahrens, *Meteorology today: an introduction to weather, climate, and the environment*. Cengage Learning, 2012. (Cited on page 22)
- [40] F. Tampieri and C. Tomasi, "Size distribution models of fog and cloud droplets in terms of the modified gamma function," *Tellus*, vol. 28, no. 4, pp. 333–347, 1976. (Cited on page 22)
- [41] D. Deirmendjian, "Scattering and Polarization Properties of Water Clouds and Hazes in the Visible and Infrared," *Appl. Opt.*, vol. 3, pp. 187–196, feb 1964. (Cited on page 23)
- [42] M. Ijaz, Z. Ghassemlooy, J. Pesek, O. Fiser, H. Le Minh, and E. Bentley, "Modeling of fog and smoke attenuation in free space optical communications link under controlled laboratory conditions," *Journal of Lightwave Technology*, 2013. (Cited on pages 23 and 24)
- [43] R. Redington, "Elements of infrared technology: Generation, transmission and detection," *Solid-State Electronics*, vol. 5, no. 5, p. 361, 1962. (Cited on pages 23 and 24)

- [44] I. I. Kim, B. McArthur, and E. J. Korevaar, “Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications,” in *Optical Wireless Communications III* (E. J. Korevaar, ed.), vol. 4214 of *SPIE Proceedings*, pp. 26–37, SPIE, 2001. (Cited on pages 23 and 24)
- [45] F. Nadeem, T. Javornik, E. Leitgeb, V. Kvicera, and G. Kandus, “Continental fog attenuation empirical relationship from measured visibility data,” *Radioengineering*, 2010. (Cited on page 24)
- [46] M. Grabner and V. Kvicera, “The wavelength dependent model of extinction in fog and haze for free space optical communication,” *Optics Express*, vol. 19, no. 4, p. 3379, 2011. (Cited on page 24)
- [47] World Meteorological Organization, “Guide to meteorological instruments and methods of observation.,” *Quality Assurance*, no. 8, p. 678, 2018. (Cited on pages 24, 25, 48, 49, 50, 53, and 59)
- [48] A. Prokes, “Atmospheric effects on availability of free space optics systems,” *Optical Engineering*, 2009. (Cited on page 25)
- [49] E. O. Hulburt, “Optics of Atmospheric Haze,” *J. Opt. Soc. Am.*, vol. 31, pp. 467–476, jul 1941. (Cited on page 26)
- [50] A. M. Society, “Rain - AMS Glossary.” <http://glossary.ametsoc.org/wiki/Rain>, 2012. (Cited on pages 26, 28, 29, and 74)
- [51] DWD, “Wetter und Klima - Deutscher Wetterdienst - Glossar - Niederschlagsintensität.” <https://www.dwd.de/DE/service/lexikon/Functions/glossar.html;jsessionid=C5D8A14B1FC260E5C6A82328C542C755.live31093?lv2=102672{&}lv3=102812>, 2020. (Cited on pages 28 and 29)
- [52] Daimler AG, “Mercedes-Benz inaugurates new climatic wind tunnels: Weather at the touch of a button.” <https://media.daimler.com/marsMediaSite/ko/de/9915555>, 2011. (Cited on pages 29 and 30)
- [53] H. Boeriu, “BMW’s Energy and Environmental Test Center.” <https://www.bmwblog.com/2010/05/21/bmws-energy-and-environmental-test-center/>, 2020. (Cited on pages 29 and 30)
- [54] M. Colomb, P. Duthon, and S. Laukkanen, “Characteristics of Adverse Weather Conditions,” *Dense*, 2017. (Cited on pages 30, 31, and 32)

- [55] Technische Hochschule Ingolstadt, “Carissma - Indoor-Versuchsanlage.” <https://www.thi.de/forschung/carissma/labore/indoor-versuchsanlage/>, 2019. (Cited on pages 30, 31, and 32)
- [56] T. Lassu, M. Seeger, P. Peters, and S. D. Keesstra, “The Wageningen Rainfall Simulator: Set-up and Calibration of an Indoor Nozzle-Type Rainfall Simulator for Soil Erosion Studies,” *Land Degradation and Development*, vol. 26, no. 6, pp. 604–612, 2015. (Cited on page 30)
- [57] T. Iserloh, J. B. Ries, J. Arnáez, C. Boix-Fayos, V. Butzen, A. Cerdà, M. T. Echeverría, J. Fernández-Gálvez, W. Fister, C. Geißler, J. A. Gómez, H. Gómez-Macpherson, N. J. Kuhn, R. Lázaro, F. J. León, M. Martínez-Mena, J. F. Martínez-Murillo, M. Marzen, M. D. Mingo-rance, L. Ortigosa, P. Peters, D. Regüés, J. D. Ruiz-Sinoga, T. Scholten, M. Seeger, A. Solé-Benet, R. Wengel, and S. Wirtz, “European small portable rainfall simulators: A comparison of rainfall characteristics,” *Catena*, vol. 110, pp. 100–112, 2013. (Cited on page 30)
- [58] A. S. Claassens and H. V. van der Watt, “An inexpensive, portable rain simulator: Construction and test data,” *South African Journal of Plant and Soil*, vol. 10, no. 1, pp. 6–11, 1993. (Cited on page 30)
- [59] M. Esteves, O. Planchon, J. M. Lapetite, N. Silvera, and P. Cadet, “The ‘EMIRE’ large rainfall simulator: Design and field testing,” in *Earth Surface Processes and Landforms*, vol. 25, pp. 681–690, 2000. (Cited on page 30)
- [60] S. C. Carvalho, J. L. de Lima, and M. I. P. de Lima, “Using meshes to change the characteristics of simulated rainfall produced by spray nozzles,” *International Soil and Water Conservation Research*, 2014. (Cited on page 30)
- [61] Y. Yamawaki, Toshiki and Yamano, Shin-ichi and Katogi, Yutaka and Tamura, Toshihito and Ohira, “Millimeter-wave obstacles detection radar,” *Fujitsu Tech*, vol. 15, pp. 1—13, 2000. (Cited on page 31)
- [62] M. Colomb, K. Hirech, P. André, J. J. Boreux, P. Lacôte, and J. Dufour, “An innovative artificial fog production device improved in the European project “FOG”,” *Atmospheric Research*, vol. 87, no. 3-4, pp. 242–251, 2008. (Cited on pages 31, 54, 73, 84, and 123)
- [63] E. Rabiei, U. Haberlandt, M. Sester, and D. Fitzner, “Rainfall estimation using moving cars as rain gauges—laboratory experiments,” *Hydrology*

- and Earth System Sciences*, vol. 17, no. 11, pp. 4701–4712, 2013. (Cited on page 31)
- [64] M. Colomb, J. Dufour, M. Hirech, P. Morange, J. J. Boreux, P. Lacôte, K. Hirech, and P. Lacote, “Innovative artificial fog production device-A technical facility for research activities,” *The third international conference on fog, fog collection and dew*, no. 1, pp. 1–5, 2004. (Cited on pages 31, 73, and 75)
- [65] T. Gibson, “Virginia’s Smart Road: Where Researchers Make the Extreme Weather,” *Weatherwise*, vol. 68, no. 4, pp. 20–27, 2015. (Cited on page 31)
- [66] C. M. Bishop, *Pattern recognition and machine learning*. Springer-Verlag New York, 2006. (Cited on pages 33, 34, 35, 36, 37, 38, 39, 42, 43, and 44)
- [67] H. Habibi Aghdam and E. Jahani Heravi, *Guide to Convolutional Neural Networks*. 2017. (Cited on pages 33, 35, 38, 40, 43, and 44)
- [68] M. Skilton, F. Hovsepian, M. Skilton, and F. Hovsepian, “The 4th Industrial Revolution Impact,” in *The 4th Industrial Revolution*, 2018. (Cited on pages 33, 34, 35, 36, 42, and 44)
- [69] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, pp. 255 LP – 260, jul 2015. (Cited on page 33)
- [70] R. de Mello and M. Antonelli Ponti, *A Brief Review on Machine Learning*, pp. 1–74. Cham: Springer International Publishing, 2018. (Cited on page 33)
- [71] Lars Fischer, “Nach nur vier Stunden üben: Künstliche Intelligenz schlägt besten Schachcomputer der Welt - Spektrum der Wissenschaft.” <https://www.spektrum.de/news/kuenstliche-intelligenz-schlaegt-besten-schachcomputer-der-welt/1524575>, 2017. (Cited on page 33)
- [72] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*, pp. 151–170. Dordrecht: Springer Netherlands, 1996. (Cited on page 33)
- [73] P. Y. Simard, D. Steinkraus, and J. Platt, “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis,” Insti-

- tute of Electrical and Electronics Engineers, Inc., aug 2003. (Cited on page 39)
- [74] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Neural Information Processing Systems*, vol. 25, 2012. (Cited on pages 40 and 52)
- [75] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML’10, (Madison, WI, USA), pp. 807–814, Omnipress, 2010. (Cited on page 40)
- [76] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, vol. 28, 2013. (Cited on pages 40, 66, and 67)
- [77] D. Kriesel, “A Brief Introduction to Neural Networks,” 2007. (Cited on pages 42 and 44)
- [78] M. A. Rahman and Y. Wang, “Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation,” in *Advances in Visual Computing* (G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, eds.), (Cham), pp. 234–244, Springer International Publishing, 2016. (Cited on page 44)
- [79] F. P. J. Piewak, *LiDAR-based Semantic Labeling : Automotive 3D Scene Understanding*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2020. (Cited on page 44)
- [80] F. Krüger, *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models*. PhD thesis, 2016. (Cited on page 45)
- [81] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes Challenge: A Retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015. (Cited on page 46)
- [82] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013. (Cited on pages 46, 53, 58, 60, 68, and 71)
- [83] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset

- for Semantic Urban Scene Understanding,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. (Cited on pages 46, 68, and 114)
- [84] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” no. iii, 2019. (Cited on pages 46 and 68)
- [85] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, “Weather Influence and Classification with Automotive Lidar Sensors,” in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1527–1534, IEEE, 2019. (Cited on pages 47, 58, 61, 72, 73, 76, 89, 96, 97, and 98)
- [86] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, “CNN-based Lidar Point Cloud De-Noising in Adverse Weather,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. (Cited on pages 47, 61, 68, 79, 81, 87, and 114)
- [87] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, “CNN-Based Lidar Point Cloud De-Noising in Adverse Weather,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 2514–2521, apr 2020. (Cited on pages 47, 61, 66, 67, 68, 71, 73, 79, 81, 87, 88, and 114)
- [88] M. W. Michael Holz, Jörg Moisel, “Verfahren und Vorrichtung zur quantitativen Abschätzung der Sichtweite in Fahrzeugen,” jan 2003. (Cited on page 48)
- [89] SICK AG, “Messung der Sichtweite | SICK.” <https://www.sick.com/de/de/branchen/verkehr/strasse/loesungen-zur-verkehrssicherheit-an-strassen/messung-der-sichtweite/c/p329714>, 2020. (Cited on page 48)
- [90] SICK AG, “Sichtweitemessgeräte | VISIC620 | SICK.” <https://www.sick.com/de/de/verkehrssensoren/sichtweitemessgeraete/visic620/c/g57494>, 2020. (Cited on page 48)
- [91] Südwestrundfunk, “Falsche Tempolimits auf der A8 zwischen Ulm und Mühlhausen - SWR Aktuell.” <https://www.swr.de/swraktuell/baden-wuerttemberg/ulm/nebelwarnung-trotz-sonnenschein-auf-der-a8-100.html>, 2020. (Cited on page 49)

- [92] Optical sensors Sweden AB, “Optical Fog Sensor - OFS.” <http://www.opticalsensors.se/of.html>, 2021. (Cited on page 49)
- [93] K. Y. Kazuki Sakata, “Sensor in car window,” 2004. (Cited on page 50)
- [94] A. U. Kuehnle, “Vehicle rain sensor,” 2002. (Cited on page 50)
- [95] S. Y. Han, “Rain sensor,” 2013. (Cited on page 50)
- [96] C. Busch and E. Debes, “Wavelet transform for analyzing fog visibility,” *IEEE Intelligent Systems and Their Applications*, 1998. (Cited on page 51)
- [97] D. Pomerleau, “Visibility estimation from a moving vehicle using the Ralph vision system,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1997. (Cited on pages 51 and 52)
- [98] N. Hautière, J. P. Tarel, H. Halmaoui, R. Brémond, and D. Aubert, “Enhanced fog detection and free-space segmentation for car navigation,” *Machine Vision and Applications*, vol. 25, no. 3, pp. 667–679, 2014. (Cited on page 52)
- [99] M. Roser and F. Moosmann, “Classification of weather situations on single color images,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 798–803, 2008. (Cited on pages 52 and 65)
- [100] M. Pavlic, G. Rigoll, and S. Ilic, “Classification of images in fog and fog-free scenes for use in vehicles,” *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 481–486, 2013. (Cited on page 52)
- [101] M. Elhoseiny, S. Huang, and A. Elgammal, “Weather classification with deep convolutional neural networks,” in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3349–3353, IEEE, 2015. (Cited on page 53)
- [102] C. Lu, D. Lin, J. Jia, and C.-K. K. Tang, “Two-Class Weather Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2510–2524, dec 2017. (Cited on page 53)
- [103] C. Lu, D. Lin, J. Jia, and C. K. Tang, “Two-Class Weather Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2510–2524, dec 2017. (Cited on pages 53 and 65)
- [104] J. Zhu, D. Dolgov, and D. Ferguson, “Methods and systems for detecting weather conditions including fog using vehicle onboard sensors,” 2015. (Cited on pages 53 and 59)

- [105] M. Al Naboulsi, “Fog attenuation prediction for optical and infrared waves,” *Optical Engineering*, vol. 43, p. 319, feb 2004. (Cited on pages 53, 59, and 62)
- [106] T. Peynot, J. Underwood, and S. Scheduling, “Towards reliable perception for unmanned ground vehicles in challenging conditions,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1170–1176, 2009. (Cited on pages 53, 58, and 62)
- [107] A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias, “Quantifying the influence of rain in LiDAR performance,” *Measurement: Journal of the International Measurement Confederation*, vol. 95, pp. 143–148, 2017. (Cited on pages 53, 54, 59, and 62)
- [108] S. Hasirlioglu, A. Riener, W. Ruber, and P. Wintersberger, “Effects of exhaust gases on laser scanner data quality at low ambient temperatures,” in *28th IEEE Intelligent Vehicles Symposium*, pp. 1708–1713, IEEE, 2017. (Cited on pages 53, 54, 59, and 62)
- [109] S. Hasirlioglu, A. Kamann, I. Doric, and T. Brandmeier, “Test methodology for rain influence on automotive surround sensors,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 2242–2247, IEEE, 2016. (Cited on pages 53, 54, 59, and 62)
- [110] S. Hasirlioglu, I. Doric, C. Lauerer, and T. Brandmeier, “Modeling and simulation of rain for the test of automotive sensor systems,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2016-Augus, pp. 286–291, IEEE, 2016. (Cited on pages 53 and 62)
- [111] M. Ijaz, Z. Ghassemlooy, H. Le Minh, S. Rajbhandari, and J. Perez, “Analysis of fog and smoke attenuation in a free space optical communication link under controlled laboratory conditions,” in *2012 International Workshop on Optical Wireless Communications (IWOW)*, pp. 1–3, 2012. (Cited on pages 53, 59, and 62)
- [112] R. H. Rasshofer, M. Spies, and H. Spies, “Influences of weather phenomena on automotive laser radar systems,” *Advances in Radio Science*, vol. 9, pp. 49–60, 2011. (Cited on pages 53, 59, and 62)
- [113] J. Ryde and N. Hillier, “Performance of laser and radar ranging devices in adverse environmental conditions,” *Journal of Field Robotics*, vol. 26, no. 9, pp. 712–727, 2009. (Cited on pages 53, 54, 55, 59, and 62)

- [114] P. Sallis, C. Dannheim, C. Icking, and M. Maeder, “Air Pollution and Fog Detection through Vehicular Sensors,” in *2014 8th Asia Modelling Symposium*, pp. 181–186, 2014. (Cited on pages 53, 59, and 62)
- [115] A. U. Shamsudin, K. Ohno, T. Westfechtel, S. Takahiro, Y. Okada, and S. Tadokoro, “Fog removal using laser beam penetration, laser intensity, and geometrical features for 3D measurements in fog-filled room,” *Advanced Robotics*, vol. 30, no. 11-12, pp. 729–743, 2016. (Cited on pages 53, 55, 57, 59, 62, and 64)
- [116] T. G. Phillips, N. Guenther, and P. R. McAree, “When the dust settles: The four behaviors of LIDAR in the presence of fine airborne particulates,” *Journal of field robotics*, vol. 34, no. 5, pp. 985–1009, 2017. (Cited on pages 53, 55, 58, 59, 62, 86, and 93)
- [117] M. Bijelic, T. Gruber, and W. Ritter, “A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 760–767, IEEE, 2018. (Cited on pages 53, 54, 58, 59, and 62)
- [118] Gaurav Pandey, James R. McBride, and Ryan M. Eustice, “Ford campus vision and lidar data set,” *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011. (Cited on pages 53, 58, and 71)
- [119] V. Sandner, “Development of a test target for AEB systems,” in *23rd international technical conference on the enhanced safety of vehicles (ESV)*, 2013. (Cited on pages 54 and 74)
- [120] J. Wojtanowski, M. Zygmunt, M. Kaszczuk, Z. Mierczyk, and M. Muzal, “Comparison of 905 nm and 1550 nm semiconductor laser rangefinders’ performance deterioration due to adverse environmental conditions,” *Opto-electronics Review*, vol. 22, no. 3, pp. 183–190, 2014. (Cited on page 55)
- [121] M. Kutila, P. Pyykonen, W. Ritter, O. Sawade, and B. Schaufele, “Automotive LIDAR sensor development scenarios for harsh weather conditions,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 265–270, 2016. (Cited on pages 55 and 59)
- [122] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb, and P. Duthon, “Automotive LiDAR performance verification in fog and rain,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1695–1701, nov 2018. (Cited on pages 55, 58, and 59)

- [123] N. Charron, S. Phillips, and S. L. Waslander, “De-Noising of Lidar Point Clouds Corrupted by Snowfall,” *2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 254–261, may 2018. (Cited on pages 55, 56, 57, 58, 60, 115, and 116)
- [124] M. Carfagni, R. Furferi, L. Governi, M. Servi, F. Uccheddu, and Y. Volpe, “On the Performance of the Intel SR300 Depth Camera: Metrological and Critical Characterization,” *IEEE Sensors Journal*, 2017. (Cited on page 56)
- [125] J. Shen and S. C. S. Cheung, “Layer depth denoising and completion for structured-light RGB-D cameras,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013. (Cited on pages 56 and 60)
- [126] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 839–846, IEEE, 1998. (Cited on pages 56 and 60)
- [127] S. Rönnbäck and Å. Wernersson, “On filtering of laser range data in snowfall,” *2008 4th International IEEE Conference Intelligent Systems, IS 2008*, vol. 3, pp. 1733–1739, 2008. (Cited on pages 56 and 60)
- [128] Y. Li, J. Li, L. Wang, J. Zhang, D. Li, and M. Zhang, “A weighted least squares algorithm for time-of-flight depth image denoising,” *Optik*, vol. 125, pp. 3283–3286, 2014. (Cited on pages 56 and 60)
- [129] O. Schall, A. Belyaev, and H. P. Seidel, “Robust filtering of noisy scattered point data,” in *Point-Based Graphics, 2005 - Eurographics/IEEE VGTC Symposium Proceedings*, 2005. (Cited on pages 56 and 66)
- [130] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, “Bayesian point cloud reconstruction,” *Computer Graphics Forum*, vol. 25, no. 3, pp. 379–388, 2006. (Cited on pages 56 and 60)
- [131] L. Chen, H. Lin, and S. Li, “Depth image enhancement for Kinect using region growing and bilateral filter,” *Proceedings - International Conference on Pattern Recognition*, no. November, pp. 3070–3073, 2012. (Cited on pages 56 and 60)
- [132] A. V. Le, S. W. Jung, and C. S. Won, “Directional joint bilateral filter for depth images,” *Sensors (Switzerland)*, vol. 14, no. 7, pp. 11362–11378, 2014. (Cited on pages 56 and 60)

- [133] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. (Cited on pages 57 and 60)
- [134] P. Sun, H. Kretschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. (Cited on pages 58, 60, and 71)
- [135] Scale AI Inc. and Hesai Photonics Technology Co. Ltd, “Pandaset by Hesai and Scale.” 2021. (Cited on pages 58, 60, and 71)
- [136] A. Djuricic and B. Jutzi, “Supporting UAVs in low visibility conditions by multiple-pulse laser scanning devices,” *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 1, p. W1, 2013. (Cited on page 58)
- [137] F. Piewak, P. Pinggera, M. Enzweiler, D. Pfeiffer, and M. Zöllner, “Improved Semantic Stixels via Multimodal Sensor Fusion,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11269 LNCS, pp. 447–458, Springer Verlag, 2019. (Cited on pages 59 and 116)
- [138] S. Hasirlioglu, I. Doric, A. Kamann, and A. Rienr, “Reproducible Fog Simulation for Testing Automotive Surround Sensors,” in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–7, 2017. (Cited on page 59)
- [139] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2Noise: Learning Image Restoration without Clean Data,” no. 3, 2018. (Cited on page 59)
- [140] W. Yang, K. Wang, and W. Zuo, “Neighborhood component feature selection for high-dimensional data,” *Journal of Computers*, 2012. (Cited on page 64)
- [141] Z. Zhang, H. Ma, H. Fu, and C. Zhang, “Scene-free multi-class weather classification on single images,” *Neurocomputing*, 2016. (Cited on page 65)
- [142] D. D. Webster and T. P. Breckon, “Improved raindrop detection using combined shape and saliency descriptors with scene context isolation,”

- in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2015-Decem, pp. 4376–4380, 2015. (Cited on page 65)
- [143] J. C. Villarreal Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, “Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks,” in *2018 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2018*, pp. 305–310, IEEE, 2018. (Cited on page 65)
- [144] L. W. Kang, K. L. Chou, and R. H. Fu, “Deep learning-based weather image recognition,” in *Proceedings - 2018 International Symposium on Computer, Consumer and Control, IS3C 2018*, 2019. (Cited on page 65)
- [145] F. Piewak, P. Pinggera, M. Schäfer, D. Peter, B. Schwarz, N. Schneider, M.ENZweiler, D. Pfeiffer, and M. Zöllner, “Boosting LIDAR-based semantic labeling by cross-modal training data generation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11134 LNCS, pp. 497–513, 2019. (Cited on pages 65, 66, 67, 68, 114, and 115)
- [146] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation,” 2019. (Cited on pages 65, 66, 67, 68, 108, 115, and 116)
- [147] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” jul 2012. (Cited on page 67)
- [148] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015. (Cited on page 67)
- [149] B. Wu, A. Wan, X. Yue, and K. Keutzer, “SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud,” oct 2017. (Cited on page 67)
- [150] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. (Cited on page 68)
- [151] B. Yang, W. Luo, and R. Urtasun, “PIXOR: Real-time 3D Object Detection from Point Clouds,” in *Proceedings of the IEEE Computer Society*

- Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, IEEE Computer Society, dec 2018. (Cited on page 68)
- [152] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast Encoders for Object Detection from Point Clouds,” dec 2018. (Cited on page 68)
- [153] B. Li, “3D fully convolutional network for vehicle detection in point cloud,” in *IEEE International Conference on Intelligent Robots and Systems*, 2017. (Cited on page 68)
- [154] G. J. Van Oldenborgh, P. Yiou, and R. Vautard, “On the roles of circulation and aerosols in the decline of mist and dense fog in Europe over the last 30 years,” *Atmospheric Chemistry and Physics*, vol. 10, no. 10, pp. 4597–4609, 2010. (Cited on page 69)
- [155] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, “Seeing Through Fog Without Seeing Fog: Deep Multi-modal Sensor Fusion in Unseen Adverse Weather,” in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2020. (Cited on pages 71, 75, 77, 81, 82, 83, and 125)
- [156] European Center for Information and Communication Technologies, “DENSE - 24/7 AUTOMOTIVE SENSING SYSTEM.” <https://dense247.eu/home/>, 2019. (Cited on pages 71, 73, 75, and 77)
- [157] T. Gruber, M. Bijelic, F. Heide, W. Ritter, and K. Dietmayer, “Pixel-Accurate Depth Evaluation in Realistic Driving Scenarios,” *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, pp. 95–105, jun 2019. (Cited on pages 71, 73, 75, and 125)
- [158] T. Gruber, F. Julca-Aguilar, M. Bijelic, and F. Heide, “Gated2Depth: Real-Time Dense Lidar From Gated Images,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1506–1516, IEEE, oct 2019. (Cited on pages 71 and 73)
- [159] M. Bijelic, P. Kysela, T. Gruber, W. Ritter, and K. Dietmayer, “Recovering the Unseen: Benchmarking the Generalization of Enhancement Methods to Real World Data in Heavy Fog,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, jun 2019. (Cited on pages 71 and 73)
- [160] F. Piewak, P. Pinggera, and M. Zöllner, “Analyzing the Cross-Sensor Portability of Neural Network Architectures for LiDAR-based Seman-

tic Labeling,” in *IEEE Intelligent Transportation Systems Conference (ITSC)*, jul 2019. (Cited on page 76)

Personal Publications

Journal Article

- [161] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. CNN-Based Lidar Point Cloud De-Noising in Adverse Weather. *IEEE Robotics and Automation Letters*, 5(2):2514–2521, apr 2020. (Cited on pages 47, 61, 66, 67, 68, 71, 73, 79, 81, 87, 88, and 114)

Conference Proceedings

- [162] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. CNN-based Lidar Point Cloud De-Noising in Adverse Weather. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. (Cited on pages 47, 61, 68, 79, 81, 87, and 114)
- [163] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather Influence and Classification with Automotive Lidar Sensors. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1527–1534. IEEE, 2019. (Cited on pages 47, 58, 61, 72, 73, 76, 89, 96, 97, and 98)

Patents

- [164] Robin Heinzler. Verfahren zum Betrieb einer Beleuchtungseinheit, Patent DE: 10 2020 000 961.1, 2020. (Not cited.)
- [165] Robin Heinzler. Verfahren zur Erfassung und Bewertung von Fahrspurmarkierungen, Patent 10 2021 004 944.6, 2021. (Not cited.)
- [166] Robin Heinzler and Florian Kraus. Verfahren zur tastengesteuerten akustischen Rückmeldung eines Fahrzeugs, Patent DE: 10 2019 002 268.8, 2019. (Not cited.)

- [167] Robin Heinzler and Philipp Schindler. Kraftfahrzeugvorrichtung und Verfahren zum Betrieb der Kraftfahrzeugvorrichtung, Patent DE: 10 2017 006 178.5, 2017. (Not cited.)
- [168] Robin Heinzler and Philipp Schindler. Verfahren zur Bestimmung einer Reichweite eines Umgebungssensors für ein Fahrzeug, Patent DE: 10 2018 003 784.4, 2018. (Not cited.)
- [169] Robin Heinzler and Philipp Schindler. Verfahren zur Wetter- und/oder Sichtweitenerkennung, Patent 10 2018 008 442.7, 2018. (Not cited.)
- [170] Robin Heinzler and Philipp Schindler. Verfahren zum Betreiben einer Erfassungsvorrichtung für ein Kraftfahrzeug, sowie Erfassungsvorrichtung, Patent DE: 10 2019 005 483.0, 2019. (Not cited.)
- [171] Robin Heinzler and Philipp Schindler. Verfahren zur Erkennung von Verkehrszeichen für ein Fahrzeug, Patent DE: 10 2019 005 583.7, 2019. (Not cited.)
- [172] Robin Heinzler and Philipp Schindler. Verfahren zur Ermittlung einer Sichtweite eines Sensors, Patent DE: 10 2018 008 903.8, 2019. (Not cited.)
- [173] Robin Heinzler and Philipp Schindler. Verfahren zur Ermittlung von atmosphärischen Störungen mittels Lidar, Patent DE: 10 2019 007 123.9, 2019. (Not cited.)
- [174] Robin Heinzler and Philipp Schindler. Verfahren zur Objekterkennung mittels Lidar, Patent DE: 10 2019 006 401.1, 2019. (Not cited.)
- [175] Robin Heinzler and Philipp Schindler. Vorrichtung und Verfahren zur Filterung von Spiegelungen in Bilddaten, Patent 10 2021 004 337.5, 2021. (Not cited.)
- [176] Robin Heinzler, Philipp Schindler, and Manuel Schäfer. Erfassungsvorrichtung für ein Fahrzeug, Patent DE: 10 2017 009 556.6, 2017. (Not cited.)
- [177] Robin Heinzler, Philipp Schindler, and Andreas Scharf. Verfahren zur Detektion von Verunreinigungen einer optischen Sensoranordnung, Patent 10 2020 119 116.2, 2020. (Not cited.)
- [178] Robin Heinzler, Philipp Schindler, and Beate Schwarz. Verfahren zum Betreiben einer optoelektronischen Erfassungseinrichtung mit zumindest zwei Sendeleistungen, sowie optoelektronische Erfassungseinrichtung, Patent DE: 10 2019 005 578.0, 2019. (Not cited.)

Supervised Student Work

- [179] Farina Glass. *Weather Classification for Automotive LiDAR Sensors*. Master thesis, Technische Hochschule Köln, 2019. (Not cited.)
- [180] Tatjana Immel. *Analyse von Wettereinflüssen auf LiDAR-Sensoren für autonomes Fahren*. Master thesis, University of Stuttgart, 2017. (Not cited.)
- [181] Felix Laukemann. *Autonomes Fahren: Reichweitenschätzung auf Basis von Lidar-Entfernungsmessungen*. Master thesis, Karlsruhe Institute of Technology, 2018. (Not cited.)
- [182] Julian Müller. *LiDAR-Reichweitenschätzungen für Automatisiertes Fahren auf Basis von Entfernungsmessungen und Landmarkendetektionen*. Master thesis, Karlsruhe Institute of Technology, 2019. (Not cited.)