Deliverable D4.1

Realization of the first version of the AI-SEE all-weather perception system: First generative and discriminative NN models

Dissemination level	**PU**
Version	**1.0**
Lead contractor	**Mercedes-Benz AG**
Due date	**30.06.2022**
Version date	**08.07.2022**

Document information

Authors

Dr. Werner Ritter – Mercedes-Benz AG

Dr. Mario Bijelic – Mercedes-Benz AG

Stefanie Walz – Mercedes-Benz AG

Dominik Scheuble – Mercedes-Benz AG

Andrea Ramazzina – Mercedes-Benz AG

Marco Introvigne – Mercedes-Benz AG

Matthias Schulze – Algolux (Germany) GmbH

Dr. Frank Julca-Aguilar – Algolux Inc.

Funding

Co-labelled PENTA and EURIPIDES2 project endorsed by EUREKA, National Funding Authorities:
Austrian Research Promotion Agency (FFG)
Business Finland
Federal Ministry of Education and Research (BMBF)
National Research Council of Canada Industrial Research Assistance Program (NRC-IRAP)

Contact

Dr. Werner Ritter
Manager Vision Enhancement Technology
Environment Perception
Mercedes-Benz AG
RD/AFU
Werk/Plant 05919 - HPC G005-BB
D-71059 Sindelfingen, Germany
Mobile +49 (160) 863 8531
werner.r.ritter@mercedes-benz.com

LEGAL DISCLAIMER

Table of contents

List of figures

List of tables

1 Executive summary

The goal of the AI-SEE project is to realize an all-weather perception system for an automatic driving system (ADS) that operates 24 hours a day, 365 days a year, even in (e.g., weather-related) poor visibility conditions.

The first important milestone of this task is to build up the first working version of the AI-SEE all-weather perception system, as a basic building block for all further developments. This deliverable describes the necessary and performed developments of hardware and software modules for this system.

2 Introduction

The goal of the AI-SEE project is to realize an all-weather perception system for an automatic driving system (ADS) that operates 24 hours a day, 365 days a year, even in (e.g., weather-related) poor visibility conditions.

The first important milestone of this task is to build up the first working version of the AI-SEE all-weather perception system, as a basic building block for all further developments. This deliverable describes the necessary and performed developments of hardware and software modules for this system. These are:

In chapter *3 Sensor Suite,* the realization and test of a portable sensor suite, which contains all sensors necessary to accomplish the task of an all-weather ADS.

In chapter *4 Simulation of Adverse Weather Sensor Data*, two simulation tools to add the disturbances of adverse weather on sensor data of a camera and LiDAR, respectively.

In chapter *5 Weather Net for Detection of Prevailing Ambient Conditions*, to detect the ambient conditions in order to adapt the sensors and the perception system to them.

In chapter *6 Optimization of the Gated Camera*, about the first step of this self-optimization by means of a self-supervised depth calibration. The goal of this work is to increase the range and accuracy of depth measurement with this camera.

In chapter *7 Robust 3D Detector for Adverse Weather based on a Gated Camera*, a novel approach to detect objects with high accuracy even in difficult visibility conditions, both at short and long range, which outperforms current state-of-the-art-methods based on conventional sensors.

3 Sensor Suite

3.1 Design of the portable reference sensor system

In order to be able to start with the software developments in an early project phase and also to be able to clarify questions regarding sensor development and design at an early stage, a portable sensor unit was designed and set up in the AI-SEE project.

This sensor unit makes it possible to carry out measurements and data acquisition campaigns geared to the target structure at an early stage of the project and to simulate the target structure on key issues. At the same time, it serves as a reference system for the target setup. The portability of the system makes it possible to dismantle the unit from the test carrier with reasonable effort, to send it by mail to distant regions for measurement campaigns, and then to quickly reassemble and use it on another test carrier. In this way, even necessary measurement campaigns overseas (e.g. weather chamber test in Japan or the USA) can be realized with reasonable effort.

The portable sensor unit consists of a modern high-resolution MIMO RADAR from Bosch, three RGB stereo camera pairs, a gated camera stereo pair, a LiDAR and a weather station, a mobile road condition sensor and an inertial measurement unit (IMU) to measure the local environmental conditions. The portable sensor setup is shown in Figure 3.1. The sensors installed in it are explained below:

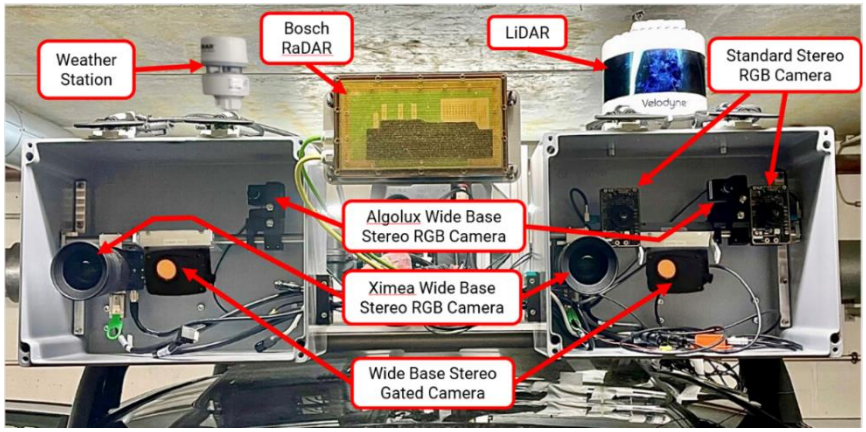

Figure 3.1: Portable sensor unit with three RGB stereo cameras, one gated stereo camera, one MIMO RADAR, one LiDAR and one weather station.

3.1.1 Stereo cameras

In total, we use three stereo camera systems for different purposes. The first is a current series system based on two on-semi AR0230 cameras with 1920 × 1024 resolution, 20 cm base width, and 12-bit quantization. The cameras run at 30 Hz and are synchronized for stereo imaging. A field of view of 39.6° is achieved with Lensagon B5M8018C optics with a focal length of 8 mm. This camera was chosen to provide data acquisition that matches the available 200 TB of the predecessor project DENSE dataset, allowing faster iteration times without troublesome domain differences. The second imaging system chosen was the on-semi AR0820 with a resolution of 3840 x 2160, a wide base width of approximately 1 m, and 12-bit quantization to show maximum capabilities. The camera runs at 15 Hz and the images are synchronized.

The mounted DSL162A lenses allow an aperture angle of 35°. Finally, a stereo Ximea CB120CG-CM-X8G3 with a wide base is installed. It provides 4096 x 3072 resolution and 12-bit quantization. The individual cameras are about 1 m apart. The optics used is a Walimex pro 35/1.4 DSLR Canon EF. This allows a 35° angle of view. The camera is operated at a frequency of 10-20 Hz. Thanks to its large sensor (APS/C), the latter camera enables the capture of high-resolution and low-noise ground images with higher photon sensitivity for simulation purposes.

3.1.2 Gated Stereo Camera

Gated images are acquired in the NIR band at 808 nm using a BrightwayVision BrightEye stereo camera operating at 120 Hz with a resolution of 1280 × 720 and a bit depth of 10 bits. The camera provides a field of view of 31.1°, similar to the stereo cameras mentioned above. Gated cameras are based on time-synchronized camera flood-lit flash laser sources [1]. The laser pulse emits a variable narrow pulse and the cameras capture the laser echo for a defined range of distances. This requires very precise camera synchronization and aperture control. This technique allows to exclude disturbances outside the depth range of interest, e.g. caused by spurious reflections from fog particles in front of the range of interest. A result image is generated by superimposing interference-reduced depth images. Compared to images from a standard camera, this is significantly reduced in interference and allows good usable images to be generated even in heavy fog, rain or snow. A depth map of the image can be generated from the overlapping distance section images (depth slices), which in terms of resolution and depth accuracy cannot currently be achieved by any other environment detection sensor used in vehicles. A new feature added in AI-SEE is the design of the gated camera as a stereo camera. This enhancement will allow to further increase the noise rejection and to significantly increase the accuracy and robustness of the depth map.

The necessary synchronization of the camera pair is not trivial, since this must be done with high precision in the nano-second range and can therefore only be achieved by a complex reprogramming of the control software on the FPGA of the camera.

3.1.3 MIMO RADAR

An "A-sample" version of Bosch's high-resolution 4D MIMO RADAR was installed for RADAR acquisition. The RADAR uses a proprietary frequency-modulated continuous-wave (FMCW) RADAR at 77 GHz with an angular resolution of 2°, an aperture angle of 100°, and ranges of up to 120 m. The RADAR provides position and velocity measurements at 10-20 Hz.

3.1.4 LiDAR

An internally rotating Velodyne VLS128 LiDAR sensor is mounted on top of the left measurement box in the direction of travel. The sensor operates in the NIR band at 905 nm and can provide two echoes (the strongest and the last) at 5-20 Hz. The sensor provides a non-uniform sampling pattern with increased density for upright objects on the horizon. The sensor has a 360° field of view and a range of up to 200 m.

3.1.5 Sensor technology to detect ambient conditions

To capture environmental conditions, we use three measurement systems in our sensor setup: we use an Airmar WX150 weather station to measure temperature, wind speed and humidity in the local

environment. To measure road conditions (water, ice, shee thickness, ground temperature, etc.) we use the Mobile Detector MD30 from Vaisala. Since this must be NEAR (<50cm) the road surface for the measurements, it is mounted separately on the low-lying towing lug of the test vehicle. To obtain accurate ego-motion data, the sensor unit is also equipped with an XSENS inertial measurement unit (IMU).

The portable sensor unit also includes a portable evaluation unit by means of which the data can be read out, processed and stored. This evaluation unit represents the vehicle computer and, like the sensor unit, can be ported to a new test vehicle at reasonable expense (e.g. for overseas trials). During the reporting period, the sensors of the portable sensor unit were also integrated into the framework of the portable evaluation unit in terms of software technology. More details can be found in Deliverable D2.2 "Individual sensor systems and interfaces".

3.2 Calibration Procedure

In order to be able to process the data of the sensors used in the reference sensor system for an event together, these must be brought to overlap in time and space in the evaluation area. The temporal overlap is usually achieved by a time synchronization of the recording times. How we realized this for our reference system is described in Deliverable D2.2 "Individual sensor systems and interfaces". To achieve spatial overlap, transformation matrices (for transforming e.g. a sensor coordinate system into the coordinate system of a neighbouring sensor or into a reference coordinate system of the vehicle) are computed in so-called sensor calibration procedures.

The calibration procedure consists of two parts - the camera intrinsic calibration, which provides the parameters for the distortion of each camera, and an extrinsic calibration which computes the positioning of the cameras (in the world coordinate system) with respect to each other.

The intrinsic calibration of the stereo cameras is performed by detecting checkerboards with pre-defined field size using the approach of [2]. An example of this process is shown in Figure 3.2. The calibration targets are placed at different positions to cover a wide range of scale, position and yaw angle to the sensor. This allows optimization of the per-camera optical distortions using the Plomb camera distortion model. The process is performed jointly for each stereo image pair. Our calibrations achieve a maximum error of about 4 pixels for a 4k stereo imager.

Figure 3.2: Intrinsic stereo calibration: The chessboard is recorded with the stereo camera at different distances and orientations with the stereo camera pair. From this information, the transformation matrix can be calculated via "least-square" method

Afterwards, the camera extrinsic can be determined. This process was implemented to calibrate each camera to the LiDAR sensor. From the calibration process we implemented, a global rigid-body transformation is obtained from each optical sensor to the LiDAR sensor. Using the TF-Tree implementation in the ROS-framework (Robot Operating System, see https://www.ros.org/), all further transformations between the optical sensors can be easily extracted.

To register the very dense LiDAR reference point clouds on the checkerboards to the camera coordinate systems (extrinsic calibration), multiple planar checkerboard targets of known size are placed in the scene. The camera positions are extracted by running classical checkerboard detection pipelines. Using the extracted coordinate pairs from the coordinated LiDAR position and the camera pixels in the images, extrinsic calibration with respect to the LiDAR coordinate system is performed by solving the n-point perspective problem using Levenberg-Marquardt nonlinear least-squares optimization [3] [4]. For the camera LiDAR calibration, six transformation parameters need to be computed, namely the roll, pitch, yaw, x, y, and z transforms. Figure 3.3 illustrates the process.

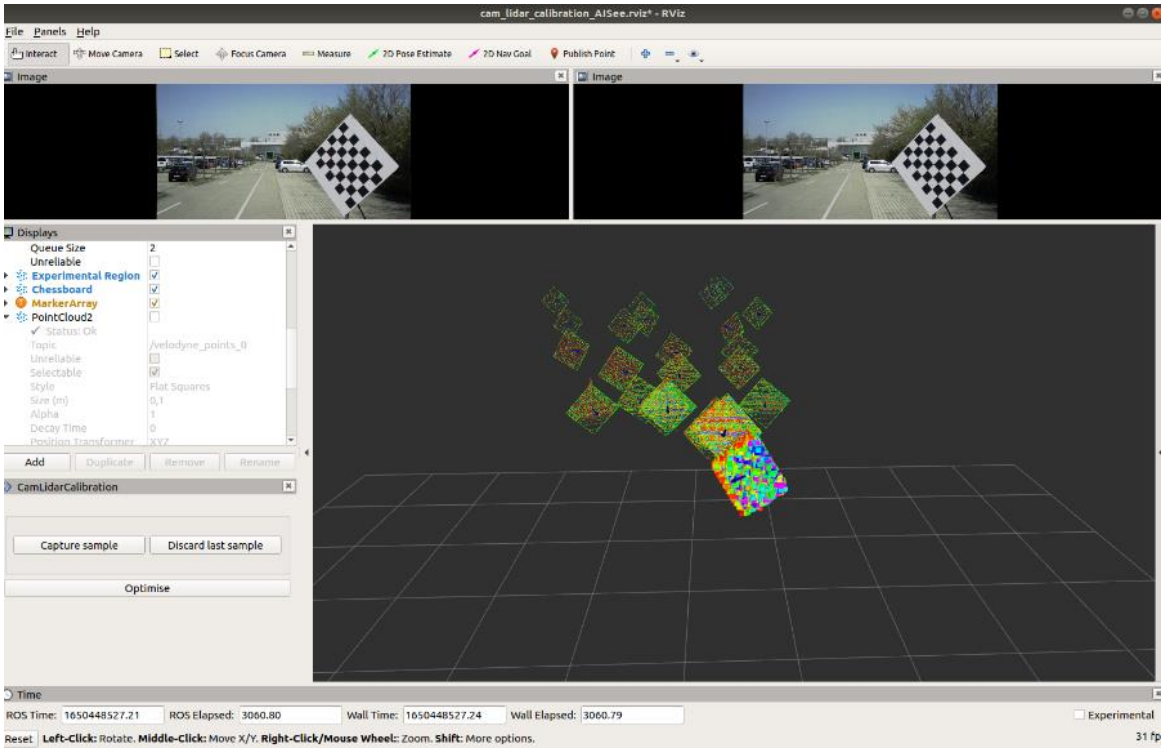

Figure 3.3: Extrinsic camera-to-LiDAR calibration pipeline. The checkerboard is detected multiple times in both the image and the LiDAR point cloud. This information is used to determine the transformation between camera and LiDAR via a least-square optimization approach.

4 Simulation of Adverse Weather Sensor Data

While adverse weather conditions that include severe scattering are heavily underrepresented in existing training and evaluation datasets [5] [6], these rare scenarios are a significant contributing factor for fatal automotive accidents [7], as a direct result of the vision impairment faced by the human drivers.

Furthermore, adverse weather conditions follow a long-tail distribution where such environment conditions are rarely encountered during day-to-day driving, making data collection, training, and evaluation challenging [8].

Due to these reasons, it is of great interest to device generative models capable of artificially creating adverse weather effects for the different input sensing modalities, such as RGB camera or LIDAR. In fact, such approaches help alleviating the data bias against adverse weather conditions, hence enabling the development of safer and more robust perception algorithms. In a nutshell, these generative models can help to increase the available training data significantly at low cost, in order to improve the performance of perception modules of ADS in real-world adverse weather scenarios.

To this end, as part of AI-SEE, we have begun developing model-based simulation methods for augmenting severe weather on sensor data, which we describe in more detail below. Please note that only these simulations allow us to develop efficient methods for better perception in adverse weather.

4.1 Fog Simulation Model for Cameras

To simulate fog into images from camera sensors, physics-based approaches have been proposed. The basic theory for these procedures lies in the Koschmieder model [9]. This model divides the effects of fog into two parts: attenuation and airlight. Attenuation describes how much light is lost due to global scattering, while airlight describes the global light intensity and the value, the light scattering converges as the distance increases.

Overall, the disturbed image I_{foggy} can be calculated from a clear weather images I_{clear} at pixel position x as follows.

$$I_{foggy}(x) = t(x)I_{clear}(x) + \big(1 - t(x)\big)L.$$

Here, $t(x)$ is the depth-dependent transmissivity and L is the global ambient component, following [10]. The transmissivity coefficient is calculated by

$$t(x) = e^{-\beta d(x)},$$

where d is the scene depth at pixel position x and β is the strength of fog scattering. Here, the fog scattering can be easily calculated from a visibility V. This visibility is defined as the distance at which 95% of all available light has been attenuated:

$$V = -\frac{ln(0.05)}{\beta}.$$

Following this model, it is possible to generate a foggy image from a clear weather one in six steps. These steps are as follows:

1. Estimation of depth from a monocular/stereo camera image for (x). Several algorithms have been proposed for predicting the depth of a scene using both classical and deep learning approaches. Some of the most recent approaches include PSMNet [11] or LEAStereo [12].
2. Refine depth by filtering inconsistent data and inferring depth anomalies.
3. Calculation of transmittance from depth: $t(x) = e^{-\beta d(x)}$.
4. Further refinement of transmission maps using a guided filter. Estimation of global airlight using the dark channel prior method [10].
5. Final application of the equation: $I_{foggy}(x) = t(x)I_{clear}(x) + \big(1 - t(x)\big)L$.

To further increase the realism of this approach, the propagation of active light sources present in the scene can be modeled in the foggy picture as well. Active light sources are, e.g., headlights from oncoming cars or traffic lights. This is done in three main steps.

First, the light sources in the scene are identified. This is currently done by applying a top-hat[1] filter to the radiance channel and filtering all objects above the target value for air illumination (calculated in the previous step 5).

Second, the propagation (in the scattering media) of these light sources is modelled. In [13], this is achieved using a simple Gaussian blurring, but more complex and accurate solutions are possible. Monte Carlo simulation is the most accurate alternative, but it is computationally expensive and impractical for large scale processing. A faster but still physically accurate alternative is to use an analytical solution for isotropic (and cone-shaped) point sources in homogeneous media. Following such an approach, the scattered light on the pixel can be computed by the integral

$$V = P \int_{\sigma} R(\mu_s, \mu_a, r, g, \boldsymbol{r}, \boldsymbol{s}`) s`^2$$

where R is the scattered radiance of an isotropic point source with a power of unity, μ_a and μ_s are respectively the absorption and scattering coefficient, g is the anisotropy factor of the phase function, and r is the module of the position of the detector. It is then possible to obtain an approximation using the average direction s of the solid angle σ (and for small distances or tenuous fog the scattered radiance can be approximated by the single scattered radiance)

$$V_a = I_{clear} 4\pi r^2 R(\mu_s, \mu_a, r, g, \boldsymbol{r}, \boldsymbol{s})\sigma$$

So that for each pixel i of the image, the light can be calculated as

$$V_{neigh,i} = \sum_{\substack{j \in lightsources \\ j \neq i}} V_a(\boldsymbol{r}_i, \boldsymbol{s}_j)$$

[1] The name Top-hat filter refers to several real-space or Fourier space filtering techniques. The name top-hat originates from the shape of the filter, which is a rectangle function, when viewed in the domain in which the filter is constructed. Source Wikipedia.

Finally, the computed light sources effect is then added to the image output from the previous step 6, hence obtaining the final fogged image

$$I_{foggy\,l}(x) = t(x)I_{clear}(x) + \big(1 - t(x)\big)L + V_{self\,ls} + V_{neigh}.$$

Figure 4.1 illustrates the clear weather reference to the artificially created fogged image. Both the improved light scattering model as well as the baseline model are shown. Clearly, the improved light scattering helps to generate a more realistic image.

Figure 4.1: Light modelling, from left to right: original (clear weather) image, fogged image (improved light scattering modelling), fogged image (baseline). By modelling the lights, the sources of illumination in the image are represented much more realistically. Instead of unnaturally heavily smeared lights represented with too large a corona (image on the right), they are now represented more sharply and in greater detail, and the red light of the taillight reflected from the road is retained (image in the middle).

With respect to other adverse weather conditions, the Koschmieder equation is also used to model the haze effect visible in rainy conditions, caused by the amount of drops that are too far away and projected onto an area smaller than one pixel. In fact, since this situation occurs as soon as the particles are a few meters away from the camera, this adverse weather effect dominates the scene. The modelling equation remains the same as before, with the only change in the attenuation coefficient given here by the precipitation rate instead of the thickness of the fog.

In addition to the rain-like fog, the individual raindrops must also be rendered with larger size. Ray casting would allow accurate modeling of the droplet photometry, but this involves very high processing costs and requires complete knowledge of the geometry and materials of the scene. This is not feasible in the real world. Therefore, this task is currently solved using a raindrop appearance database and an environment map around each drop. Finally, once the position and photometry are computed, each droplet is inserted into the image, also taking into account the defocus effect and exposure time of the camera.

With the development of the improved fog simulation, we have taken a first step towards improved perception of road users in fog. An evaluation to what extent we can achieve an improvement of the detection results is the goal of future work.

4.2 Spray Simulation Models for LiDAR

Currently, object detectors based on LiDAR rank first among different datasets [14] [15]. Compared to other sensors however, LiDAR sensors suffer severely from adverse weather affects, see, e.g. [16] [17]. In particular, wet roads lead to significant problems in LiDAR-based object detections. The reason for this is twofold. First, a water film on the road will cause an attenuated laser echo. Thus, the resulting point cloud has only a limited number of ground points, which directly affects the performance of the object detector.

Second, the wet road will cause spray from other vehicles. The laser pulse is backscattered by whirled up water droplets that are part of the spray plumes coming from other vehicles. This is especially problematic for highway driving scenarios, where the ego vehicle follows another vehicle with high velocity. Such scenarios will cause point clouds with heavy spray artefacts. The object detector subsequently mistakes these artefacts as vehicles resulting in a large number of false positives.

Similar to the previously described methods for cameras, wet road effects can be simulated and subsequently augmented onto clear weather data. This augmentation can then be used during training to improve performance in real-world wet road scenarios. In the following, the methods for simulating missing ground points and spray affects are discussed.

4.2.1 Missing ground points

As mentioned previously, the water film on the ground leads to heavily attenuated laser echos from the ground. In other words, the measured intensity of the ground points is too small for being differentiated from sensor noise. For modelling the attenuated intensities, we rely on a geometrical optical model that considers individual rays emitted by the LiDAR. From Figure 4.2, an intuitive explanation for the attenuated intensities can be gained. When the emitted ray strikes the surface of the water film, part of the ray is reflected into the air, the other part is transmitted into the water film. Inside the water film, the ray is repeatedly reflected on the surface of air and water film as well as water film and ground. Consequently, multiple increasingly attenuated rays traverse back to the sensor. The intensity registered at the sensor is then determined from the sum of all received rays.

The process of transmission and reflection can be described physically accurate by using a combination of Snell's Law and Fresnel equations. Using Snell's law, the output angle α_{out} can be computed with

$$sin\,\alpha_{out} = \frac{n_{in}}{n_{out}} sin\,\alpha_{in},$$

where n_{out} and n_{in} are the respective refractive indices, and α_{in} is the angle of the emitted laser ray. The computed angle is then used within the Fresnel equations to determine the relative amount T_{total} of the emitted power P_T traversing back to the sensor. For a detailed derivation the reader is referred to [17]. In a nutshell however, the Fresnel equations describe the relation between the amounts of reflected and transmitted power. This can be leveraged to compute T_{total} as

$$T_{total} = \frac{T_{air}\rho_0 T_{water}}{1-\rho_0 R_{water}},$$

where $T_{air}, T_{water}, R_{water}, \rho_0$ are transmitted power in air and water, reflected water, and road reflectivity in dry conditions, respectively.

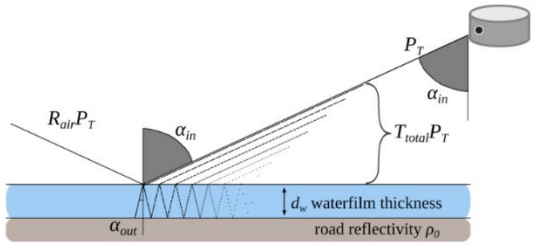

Figure 4.2: Geometrical Optical Model for modelling wet ground points

Based on these relationships, an algorithm for simulating missing ground points is designed. From a high level perspective, the algorithm can be grouped in 3 steps. First, the points in the dry reference point cloud that belong to the ground are identified. For this, we rely on a ground plane assumption and estimate points that belong to the ground using the RANSAC algorithm [18]. Second, the "wet" intensities of the identified ground points are calculated. This can be done as a function of T_{total} and the "dry" intensities from the dry reference point cloud. Finally, only points above a certain noise threshold are kept.

A qualitative result is presented in Figure 4.3, where a dry reference point cloud is compared to a point cloud captured from a real wet road and a simulated wet road. It can be seen that the point cloud from a simulated wet road matches well with the real wet road one.

4.2.2 Spray Simulation

For simulating spray, we rely on an empirical model that is based on an extensive data analysis. This model incorporates two general observations.

1. Spray points are not uniformly distributed, but form clusters
2. The number of spray points depends on object velocity, object class, distance to object, and water film height.

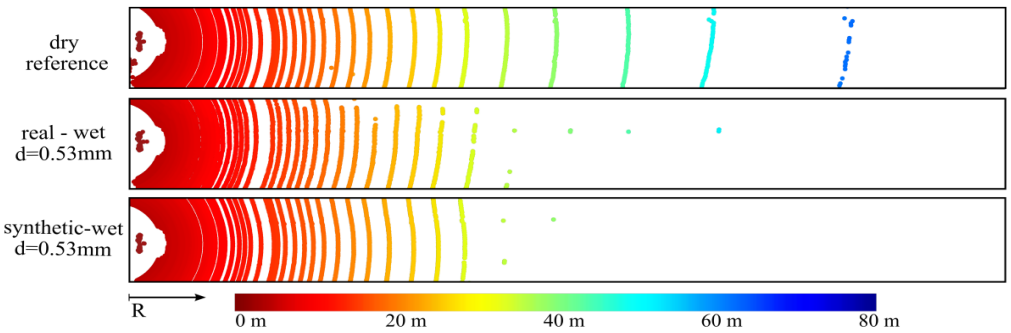

Figure 4.3: Point clouds from a dry test track (top), a watered test track (middle) and simulated wet road (bottom).

Intuitively, it can be seen that these hypotheses are true, as visualized, e.g., in Figure 4.4. However, an extensive number of experiments on a test track were conducted to validate the hypotheses and the subsequent model. For testing the first hypothesis, the DBSCAN[2] clustering algorithm is applied to point clouds from scenarios, where the ego vehicle is following vehicles of different sizes and different velocities. It was found that over 80% of the spray points can be sorted into clusters, as, e.g., also true for the example given in Figure 4.3. Due to several experiments with a variety of different sized cars at different velocities, the second hypothesis was also confirmed. However, more detailed results would be beyond the scope of this chapter.

[2] DBSCAN is a data mining algorithm for cluster analysis developed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. It is one of the most cited algorithms in the field. The algorithm works density-based and is able to detect multiple clusters. Wikipedia

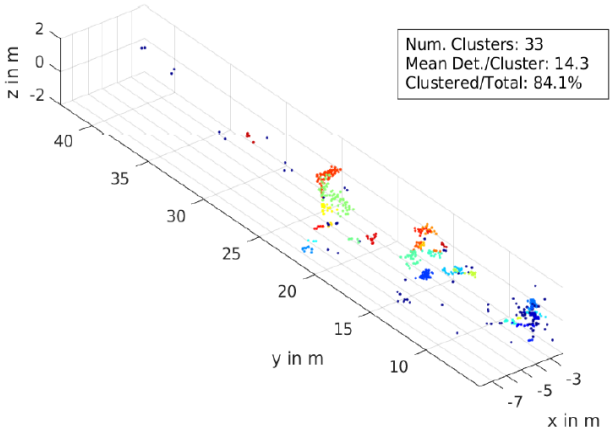

Figure 4.4: Real point cloud captured from a vehicle in 50m distance with an ego-vehicle velocity of 100 km/h

The hypotheses are subsequently used to design a spray simulation method. This method is schematically illustrated in Figure 4.5. As input, the point cloud as well as positions and velocity of all ground truth objects, e.g. other vehicles, are required. Based on this, clusters are generated. The clusters are assumed to be spherical. The cluster radius is sampled from a lognormal distribution that was fitted to real-world measurements. The number of newly generated clusters is a random process depending on the water film height and the velocity of the ground truth vehicle. The generated clusters are then appended to a global vector that is keeping track of every generated clusters. After appending new clusters, the already existing clusters are updated. This implies removing clusters with an age above a certain threshold. Furthermore, their initial velocity and the corresponding change in position is considered. Additionally, effects from wind are considered. The updated list of clusters is then used to generate the spray artefacts in the dry reference point cloud. For this, it is checked if a beam intersects with a spray cluster. With a certain probability, a point is added to the location of the closest intersection of a beam and cluster. Clearly, the intensity of this point can vary significantly. Thus, the intensity of the dry reference point belonging to this beam is attenuated in order to assign a realistic intensity to spray points.

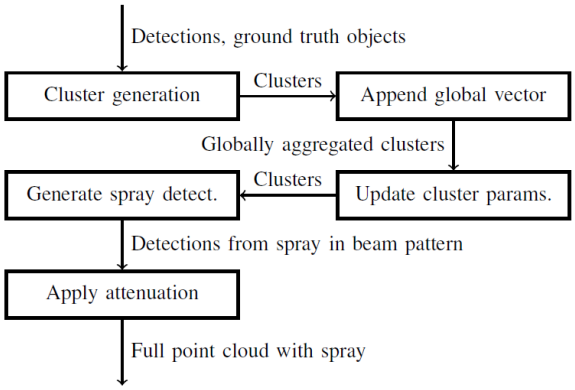

Figure 4.5: Structure of spray simulation. Point cloud and ground truth objects are inputs for every time step. A list of aggregated spray clusters is generated over time. Based on beam interactions with these clusters, the augmented spray point cloud is outputted.

The output of the simulation method is illustrated in Figure 4.6. Here, a point cloud affected by real-world spray is compared to a point cloud outputted by the spray simulation method. It can be seen that the simulated spray closely approximates the real-world spray. Especially, the generated clusters match well with the real-world reference.

Currently, we are using this spray simulation to make LiDAR-based detection techniques more robust to spray impacts. Initial results are very promising. We expect to be able to present reliable results in the near future, for example in the next deliverable D4.2.

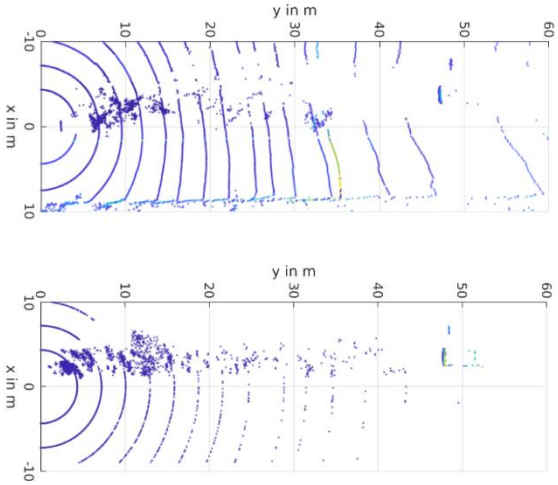

Figure 4.6: Comparison of point cloud affected by real-world spray (top) with a point cloud augmented with simulated spray (bottom).

5 Weather Net for Detection of Prevailing Ambient Conditions

It is essential for an automated driving system (ADS) to be able to recognize environmental conditions such as time of day, weather, road conditions, and road infrastructure type (highway, suburban, urban) in order to either adapt to them or, in the event of a possible overload, to delegate the driving task back to the driver in good time.

In AI-SEE, we are developing a deep neural network for this purpose, which we call Weather Net for short. The data preparation, the learning model, and the quality evaluation of this weather network that we developed are described in the following:

5.1 Dataset Preparation

First, we refine the dataset provided in [8] from the predecessor project DENSE. The dataset covers different weather and illumination scenarios and was acquired by over 10,000 km of driving in Northern Europe providing 12,997 annotated key frames. An easy extension is to also include the neighboring frames providing the same ambient weather conditions leading to 120,000 frames increase dataset size by an order of magnitude.

This dataset already included a fine-grained annotation for day light, road condition and scene setting, and ambient weather (e.g. fog, rain, snow). However, these annotations were not yet fine enough to produce satisfactory classification results for our weather network.

To achieve better environment classification results, the granularity of the annotation had to be refined according to the label hierarchy shown in Figure 5.1, into now seven categories. This avoided previously occurring misclassifications and ambiguities in the underlying data. In particular, the weather category has been divided into precipitation and fog. This is due to the fact that individual snow, rain, or hail particles are only visible over a limited distance, so they cannot be distinguished above a distance of 8 m due to the limited camera resolution. Therefore, in most images, obstacles at great distances are dominated by fog-like effects.

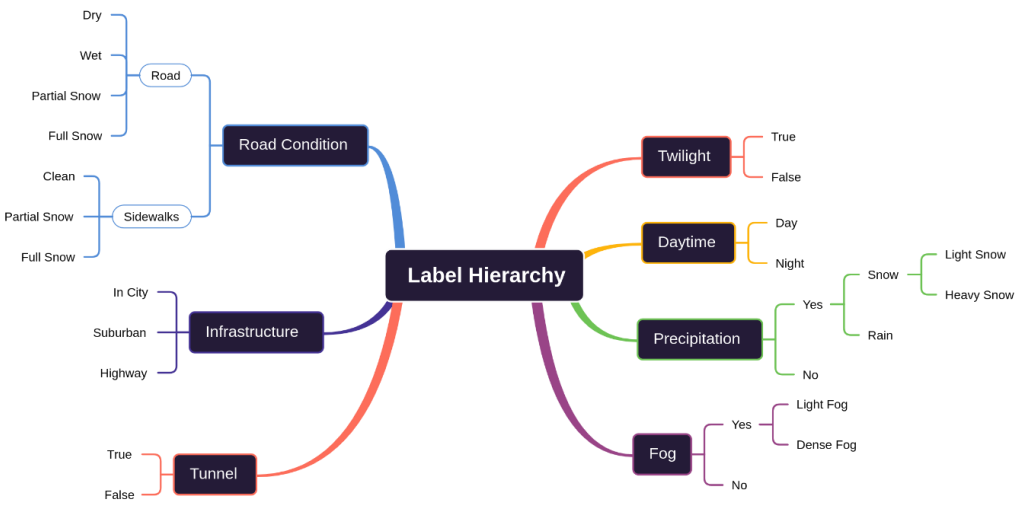

Figure 5.1: Newly proposed label hierarchy for weather classification.

The Daytime class was split into the Daytime and Twilight class because the change in illuminance is a continuous process with an intervening twilight state. When introducing the twilight state, this translation can be accounted for by considering dawn or sunrise. The Tunnel class was separated from the Infrastructure class because it was no longer possible to distinguish between city, suburbs, or highways for tunnel exits. Road condition sometimes leads to ambiguities, as previously only the drivable way was annotated without considering the sidewalks. However, snow-covered sidewalks can confuse the network when the drivable path is dry. To alleviate this problem, the condition of the sidewalk was additionally annotated.

5.2 Learning Model

Next, the deep learning model, learning the prediction of the presented label hierarchy, is introduced. It is based on a multi-output classification task. The framework is visualized in Figure 5.2. It is based on a single backbone using the EfficientNet architecture [19] and six different fully-connected-layers (FC-layers) to recognize weather and visual conditions from street-level images. The joint EfficientNet backbone enables sharing similar features between different classification tasks, reducing computation time and enables to learn feature richer representations as the learning is supervised by multiple classification tasks. Features for road coverage might be equally important for the recognition of the scattering media. This is unlike previous work [20] where independent neural networks with large backbones, e.g., ResNet50, were learned for a coarse weather classification task.

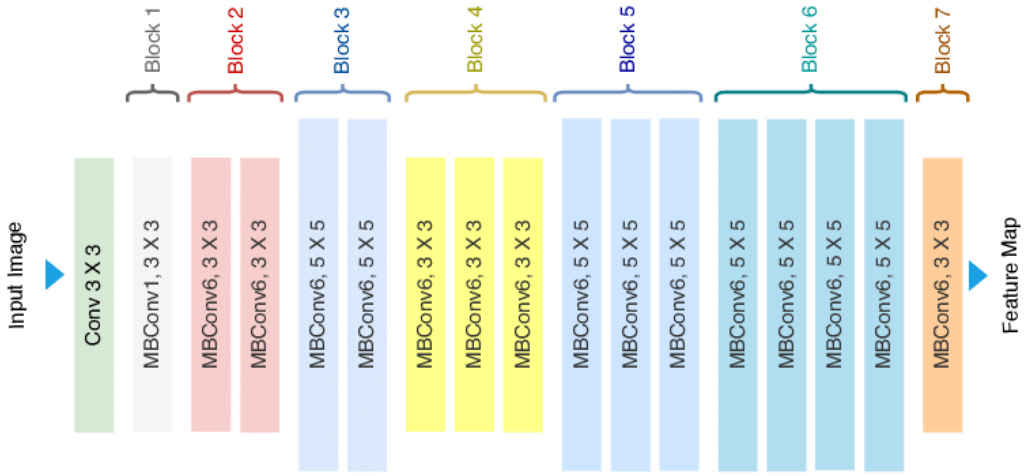

Figure 5.2: Used EfficentNet architecture for weather detection.

The main building block of this network consists of MBConv to which squeeze-and-excitation optimization is added. MBConv is similar to the inverted residual blocks used in MobileNet v2 [21]. These form a shortcut connection between the beginning and end of a convolutional block. The input activation maps are first expanded using 1x1 convolutions to increase the depth of the feature maps. This is followed by 3x3 Depth-wise convolutions and point-wise convolutions that reduce the number of channels in the output feature map. The shortcut connections connect the narrow layers whilst the wider layers are present between the skip connections. This structure helps in decreasing the overall required number of operations and the total model size.

Each FC-layer is instead composed of two dense layers where the number of neurons in the last output layer depends on the number of classes, for six different outputs.

The framework takes single-images as input. It does not require pre-defined constraints such as the camera angle, area of interest, etc.

Due to the usage of this framework, with an order of magnitude of few parameters, the computational cost is dramatically reduced. Consequently, the proposed framework can be effectively implemented in real-time environments to provide decisions on demand for autonomous vehicles with a quick and precise detection capacity.

The whole architecture is trained based on the backpropagation of error with the Adam optimizer [22]. Data augmentation techniques have been applied to enhance the training of each model. The dataset is augmented by rescaling, shearing, horizontal flips, and zooming. These techniques are commonly used to avoid overfitting and improve model performance.

Figure 5.3 shows some examples of the outputs generated by the neural network. Future implementations should use time series images in combination with data obtained from LiDAR to increase the accuracy of detection of individual snow or rain particles. This is because through the LiDAR point cloud it is possible to see particles that are not visible through RGB cameras.

Figure 5.3: Qualitative results of weather classification network

5.3 Evaluation of the Weather Net

To evaluate the multiclass classifier, we use the five well-known metrics:

- Accuracy
- Recall
- Precision
- F1-Score (is the geometric mean of precision and recall).
- AUPRC (Area under precision-recall curve or average precision)

In this way, the performance of different models can be compared or the behaviour of a model can be studied by changing the hyper parameter values. Based on a test dataset of 36,000 images, we obtained the promising evaluation shown in Table 5.1.

Table 5.1: Five metrics for the evaluation of the quality of the weather net.

CATEGORY	ACCURACY	PRECISION	RACALL	F1-SCORE	AUPRC
Daytime	0.99	0.99	0.99	0.99	0.99
Precipitation	0.89	0.88	0.89	0.88	0.89
Fog	0.93	0.93	0.93	0.93	0.97
Road Condition	0.84	0.84	0.84	0.84	0.93
Roadside Condition	0.85	0.83	0.85	0.83	0.89
Scene Setting	0.92	0.92	0.92	0.92	0.95

6 Optimization of the Gated Camera

In the previous DENSE project, the gated camera proved to be the most robust in adverse weather of all the cameras and LiDARs investigated. Even under adverse weather conditions, clear images can still be generated with a gated camera and, in addition, a pixel-precise depth maps can be calculated for these images, in a quality which are currently not achieved by any LiDAR in terms of resolution. Because of its high potential, it is worthwhile to further improve the gated camera. The goal in AI-SEE is that the gated camera can independently (self-supervised adaptivity) make the optimal settings of all parameters so that it performs optimally under the prevailing environmental conditions. A first step in this direction is the Self-Supervised Gated Depth Estimation we have developed, i.e., in a sense, the automatic self-calibration of the gated camera with respect to depth measurement.

Before discussing this Self-Supervised Gated Depth Estimation, we explain for a better understanding of this method the elementary components of a gated imaging system and the theoretical principals behind the sensing technology. Furthermore, we present approaches for deriving depth maps from multiple gated images and give overview of the corresponding ongoing work our Self-Supervised Gated Depth Estimation approach.

6.1 Principle of Gated Imaging

A gated camera is an active imaging system that consists of an amplitude-modulated source flood illuminator and a synchronized camera with gated exposure. In contrast to a LiDAR system, which directly measures the Time-of-Flight (ToF) by finding the peak of the reflected laser pulse, a gated camera system only integrates a part (depth slice) of the laser pulse. While finding the peak of a reflected laser pulse requires a very fast and sensitive sensor, a gated camera is a cost-sensitive standard image sensor with a delayed exposure that estimates the ToF by using multiple integrations with varying delay between laser illumination and camera exposure. The synchronization of the laser illumination and the exposure time of the gated camera, allow integrating only photons from a certain depth range in the scene on the CMOS imager, so called depth slices. Disturbing photons outside this depth range, e.g. caused by reflections of the laser light on fog particles, are not recorded and suppressed in this way. Thus, in contrast to standard RGB cameras and LiDARs, clear and undisturbed images can still be obtained even in foggy environments (see Figure 6.1). Another advantage of the gated camera is that a neural network can be used to generate a high-resolution pixel-precise depth map from the overlapping depth slices of a scene, as shown in Figure 6.2.

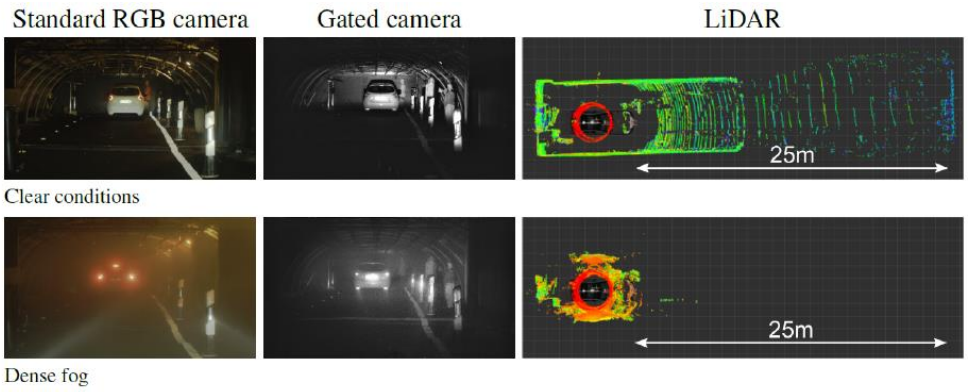

Figure 6.1: Comparison of standard RGB, gated imaging, and LiDAR sensing in clean and foggy conditions. While pedestrians are still clearly visible in the image of the gated camera, either the RGB camera or the LiDAR does not perceive them.

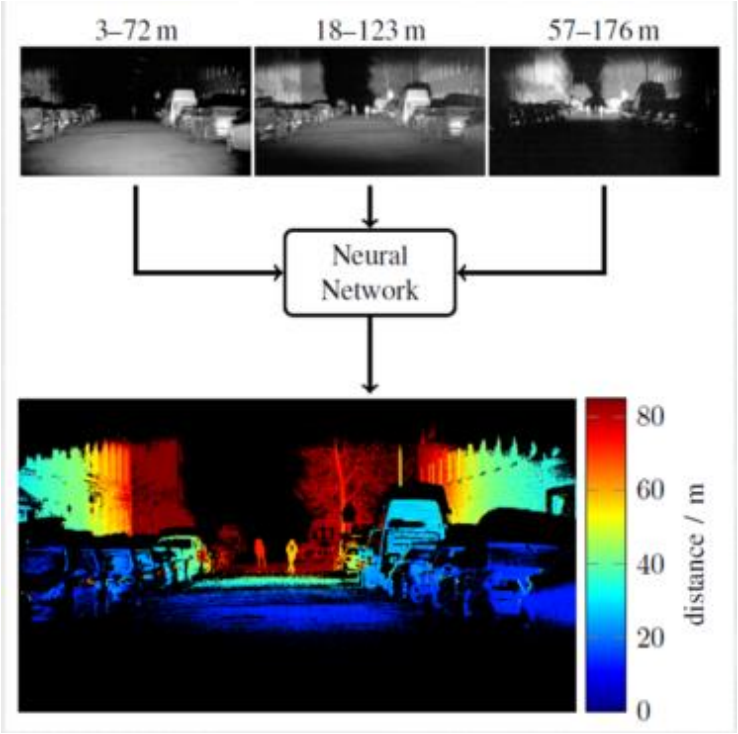

Figure 6.2: With the RangeIntensityProfiles of the overlapping areas, we can calculate a depth image with an accuracy of 5 % using a neural network.

The underlying physical conditions that lead to these properties of a gated camera are explained in more detail in the following:

The distance-dependent intensity values of a gated image can be described by so-called range-intensity-profiles (RIP). These profiles result from the convolution of the temporally modulated camera gate g and the laser pulse profile p.

We assume a rectangular laser pulse function $p(t)$ with duration t_L and rectangular gating function $g(t)$ with duration t_G. Considering a single pixel, which captures reflected photons of a point at a certain

distance r, the corresponding photons require a round-trip time of $\frac{2r}{c_0}$ to reach the camera after being emitted by the source. This means we receive the signal $p(t-\frac{2r}{c_0})$ at the sensor. The shutter of the sensor opens after a delay of ξ and remains open for the gate duration t_G. During the gating time t_G, all incident photons get integrated on the CMOS sensor. As such, the intensity value $Z(r)$ of the considered pixel is defined by the convolution of the gate pulse $g(t-\xi)$ and laser pulse $p\left(t-\frac{2r}{c_0}\right)$:

$$Z(r) = \alpha C(r) = \Phi\iota \int_{-\infty}^{\infty} g(t-\xi)p(t-\frac{2r}{c_0})\beta(r)dt$$

where Φ denotes the reflectivity, and the laser illumination ι defines the maximum amplitude of the laser pulse. The reflectivity Φ depends on the spectral distribution of the scene illumination, the reflectance of the scene surfaces, and the atmosphere's water vapor content. The atmospheric effects, which are independent of object surfaces, are modeled by

$$\beta(r) = \frac{P_{laser}\,\tau_{optics}}{4\pi r^2\, tan\left(\frac{\theta_H}{2}\right) tan\left(\frac{\theta_V}{2}\right)} \frac{\rho^2}{F_{num}^2} \frac{\lambda}{hc_0} e^{-2\gamma r}$$

with laser power $\mathrm{P_{laser}}$, horizontal/vertical field of illumination θ_H/θ_V , pixel pitch ρ, aperture F_{num}^2, wavelength λ, Planck constant h, optical transmission τ_{optics}, and atmospheric attenuation coefficient γ.

During daytime, this model is incomplete due the high spectral solar power within the NIR band that leads to a significant number of unmodulated photons captured as an ambient light Λ component. Thus, the equation from above gets extended to:

$$Z(r) = \Phi\iota \int_{-\infty}^{\infty} g(t-\xi)p\left(t-\frac{2r}{c}\right)\beta(r)dt + \Phi\kappa \int_{-\infty}^{\infty} g(t-\xi)dt$$

$$= \alpha C(r) + \Lambda$$

where κ denotes the ambient light falling on the considered point and $\Phi\kappa$ indicates the level of reflected light reaching the sensor. Assuming constant ambient light during the gating time t_G, the captured ambient light results in $\Lambda = \Phi\kappa \int_{-\infty}^{\infty} g(t-\xi)dt$.

After read-out, the final measurement $Z(r)$ for each pixel location is obtained by

$$Z(r) = \alpha C(r) + \Lambda + \eta_g + \eta_p$$

where η_p models the signal-dependent Poisson photon shot noise and η_g Gaussian read-out noise [23]. To increase the SNR, multiple laser pulses are integrated on the sensor before read-out. Three example RIPs covering different ranges are visualized in Figure 6.3.

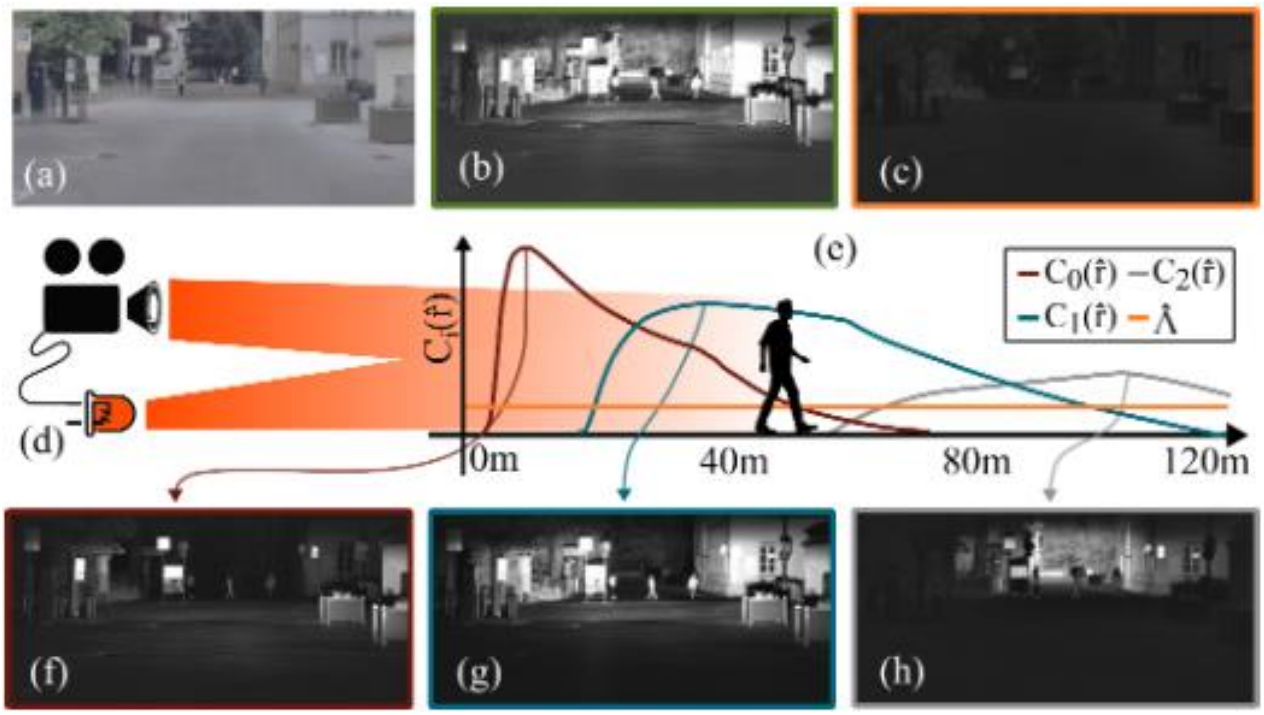

Figure 6.3: A gated camera consists of a synchronized gated camera and a flash pulsed illumination source (d). Using different exposure gates, the image formation can be described with different range-intensity profiles Ci, which are plotted depending on distance r. An overlay of all exposures is visualized in (b) and individual range intensity profiles are shown in (f), (g), and (h). Ambient light component is shown in (c) and a corresponding RGB capture of the scene is illustrated in (a).

6.2 3D Scene Reconstruction with Gated Images

This section describes several state-of-the-art methods for 3D scene reconstruction with gated images. From literature, four main approaches can be distinguished, namely time slicing [24] [25] , super-resolution depth mapping [26] [27], gain modulation [28] [29], and neural network based methods [30] [31].

6.3 Time Slicing

The time slicing method introduced by Busck et al. [25] [24] is based on so-called gated delay profiles (GDP). In contrast to the RIPs, the GDPs describe the pixel intensity of an object at a certain distance r for varying delay ξ between pulse emission and exposure time of the camera.

For the time slicing method, a sequence of N images I_i for $i \in 1,2,\dots,N$ with gate delay increased from ξ_0 in steps $\Delta\xi$ has to be recorded, resulting in a sampled version of the GDP. The depth $\hat{r}$ of an object can be computed by a conventional weighted average method

$$\hat{r} = \frac{c_0}{2}\left(\Delta\xi + \Delta\xi \frac{\sum_i i I_i}{\sum_i I_i} + \frac{t_G}{2}\right)$$

where t_G is the gate duration.

The time-slicing method requires a Gaussian laser pulse shape and a large number of gated images for sampling a small range at close distances. Increasing the range of depth estimation would require many more images or else result in lower accuracy. Therefore, high accuracy and large depth of field cannot be realized simultaneously.

6.3.1 Super-resolution depth mapping

Since time slicing requires a large number of images for sampling the gated delay profile, super-resolution methods has been introduced by Laurenzis et al. [26] [27]. This method relies on trapezoidal RIPs with equidistant rising, plateau, and falling sections that systematically overlap, see Figure 6.4. By comparing the intensity of the plateau $I_{plateau,i}$ of image i with the intensity within the linear rising ramp $I_{rising,i+1}$ of the following image $i+1$, the depth $\hat{r}$ can be estimated by

$$\hat{r} = r_0 + \frac{I_{rising,i+1}}{I_{plateau,i}} \Delta r$$

With r_0 being the start of the first slice and $\Delta r = \frac{c_0 \Delta \xi}{2}$ the scanning step given by the delay step $\Delta \xi$. The same equation can be set up for the intensity of a linear falling ramp $I_{falling,i-1}$ of the previous image $i-1$ by

$$\hat{r} = r_0 + \frac{I_{falling,i-1}}{I_{plateau,i}} \Delta r.$$

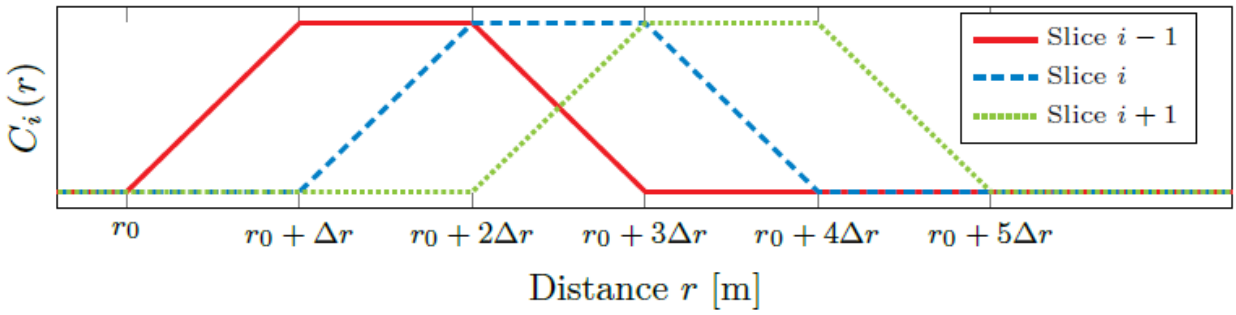

Figure 6.4: Example of the super-resolution method with three gated images shifted by Δr.

6.4 Gain Modulation

Gain modulation relies on two gated images, one with constant gain $g_1(t) = G_1 g(t)$ and one with linearly increasing gain $g_2(t) = (G_2 + kt)\, g(t)$ with linear coefficient k. The depth $\hat{r}$ for each pixel can be calculated from two images I_1 and I_2 with delay ξ by

$$\hat{r} = r_0 + \alpha\left(\frac{I_1}{I_2} - \beta\right)$$

where $r_0 = \frac{c_0}{2}(\xi + t_l)$. The coefficients α and β can be calibrated from at least two targets with known distances.

6.5 Neural network based depth estimation from gated images

Accurate depth recovery from gated images requires the recording of high-contrast images at every distance. For this purpose, the gated settings should be adapted flexibly to any scene. For example, more photons should be captured when recording more distant areas since the reflected energy of the laser pulse decreases quadratically with the distance. Hence, the gate duration should be increased for further ranges to provide high contrast images even at long distances. However, the flexible adaption of the gated settings is not feasible with the pre-mentioned methods since they have pre-defined conditions for pulse shape, gate shape, or delay times.

Introducing neural networks enables unrestricted depth reconstruction from gated images with arbitrarily modulated gated profiles. Therefore, the neural networks are trained to learn the mapping between intensity values of different gated slices and a depth value, independent of the gated settings. Figure 6.5 shows that this can be accomplished either pixel-based [31] or image-based [30].

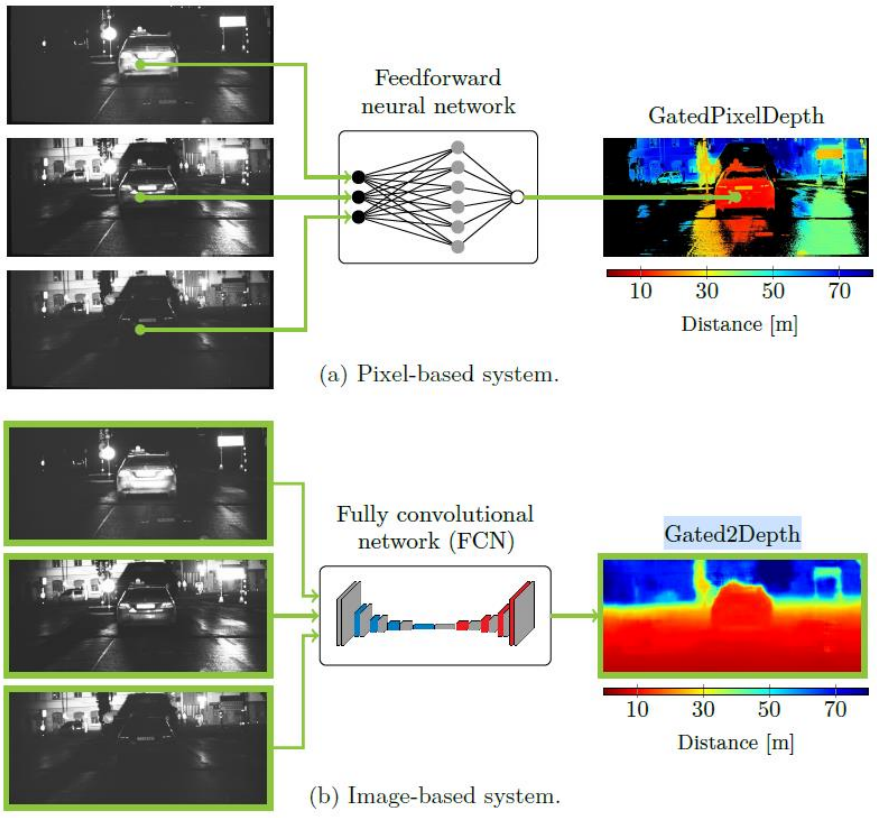

Figure 6.5: Comparison of a pixel-based and image-based system for gated depth estimation. The pixel-based network estimates the depth from a set of three pixels, while the image-based network derives the depth from three complete gated images.

6.6 Pixel-based gated depth estimation

Gruber et al. [30] was the first to take advantage of neural networks for learning the RIPs of given gated images in order to map the intensity values of different images to the corresponding depth. The proposed

neural network receives the pixel intensity values of three gated images, captured with overlapping RIPs, as input and outputs the corresponding depth value. The depth supervision of the learning process is provided by a LiDAR system. A simple multilayer neural network constitutes the fundamental of this method. Since the learning process of the neural network is pixel-based, saturated and non-illuminated pixels must be filtered out for the training and the evaluation, as they do not provide any depth information.

6.6.1 Image-based gated depth estimation

In contrast to pixel-based gated depth estimation, image-based methods use convolutional-neural networks and complete gated images as input to predict dense depth maps. The main advantage of image-based approaches is that pixels are not considered individually and semantic context across the gated images is taken into account. This facilitates the depth prediction of saturated and non-illuminated pixels. The learning process of the neural network is supervised by LiDAR measurements.

6.7 Self-Supervised Gated Depth Estimation

Previous neural network based gated depth estimation methods require LiDAR measurements for supervising the learning process. However, training neural network with sparse LiDAR ground truth depth is a challenging task since the loss function gets only evaluated at the individual LiDAR points, meaning that wrong predictions for unlabelled pixels are not penalized. This does not have that strong impact in terms of uniformly distributed sparsity labels since all pixels are still trained equally after a sufficient number of iterations. However, LiDAR measurements exhibit horizontal structures that occur due to the rotating scanning setup. Consequently, pixels lying on these horizontal lines are trained more often than pixels in between, which results in horizontal stripes in the predicted depth maps. This occurrence is shown in Figure 6.6. Furthermore, gated depth estimation supervised by LiDAR systems are limited to clear weather conditions. The reason for this is that LiDAR measurements suffer from backscatter in adverse weather. This backscatter falsifies the measurements and is visualized in Figure 6.6. Another disadvantage of ground truth LiDAR systems is that they only provide reliable depth measurements for ranges up to 100m, meaning that the evaluation of the depth estimates of the gated images is not feasible for far distances.

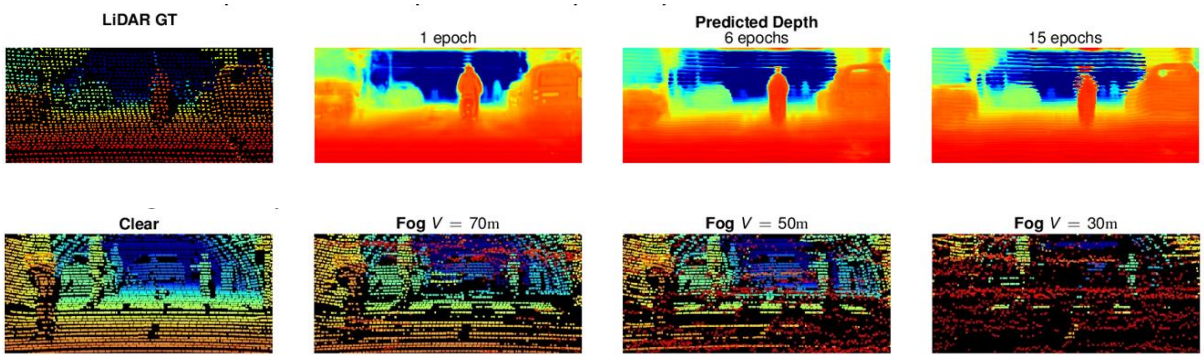

Figure 6.6: Problems of using LiDAR systems as ground truth supervision for training neural networks: Low spatial resolution of the LiDAR system causes horizontal pattern in the predicted depth maps (top), LiDAR systems suffer from backscatter in adverse weather conditions, which falsifies the ground truth measurements (bottom).

This is the reason why we want to develop a self-supervised gated depth estimation approach within the AI-SEE project that does not require any ground truth sensor for supervision. The main idea of this approach is to train a neural network self-supervised by reconstructing the input gated images with the known RIPs. Figure 6.7 visualizes the basic principle of the self-supervised network training. The encoder-decoder convolutional neural network is trained to predict simultaneously depth $\hat{r}$, albedo $\hat{\alpha}$, and ambient illumination $\widehat{\Lambda}$ from a set of three gated images $Z = [z_1, z_2, z_3]$ with overlapping RIPs. These predictions are required to reconstruct the input with the aid of the RIPs, defined by

$$Z(r) = \Phi\iota \int_{-\infty}^{\infty} g(t-\xi) p\left(t - \frac{2r}{c}\right) \beta(r) dt + \Phi\kappa \int_{-\infty}^{\infty} g(t-\xi) dt$$

$$= \alpha C(r) + \Lambda$$

Thus, a single gated image with index i can be simulated by

$$\widehat{z_\iota} = \hat{\alpha} C_i(\hat{r}) + \widehat{\Lambda}.$$

Thereby, the functions $C_i(r)$ with $j = 1,2,3$ are measured experimentally with calibrated targets and approximated with Chebyshev polynomials T_n

$$T_0 = 1, \qquad T_1 = x, \qquad T_{n+1} = 2xT_n - T_{n-1}$$

up to order of $N = 6$.

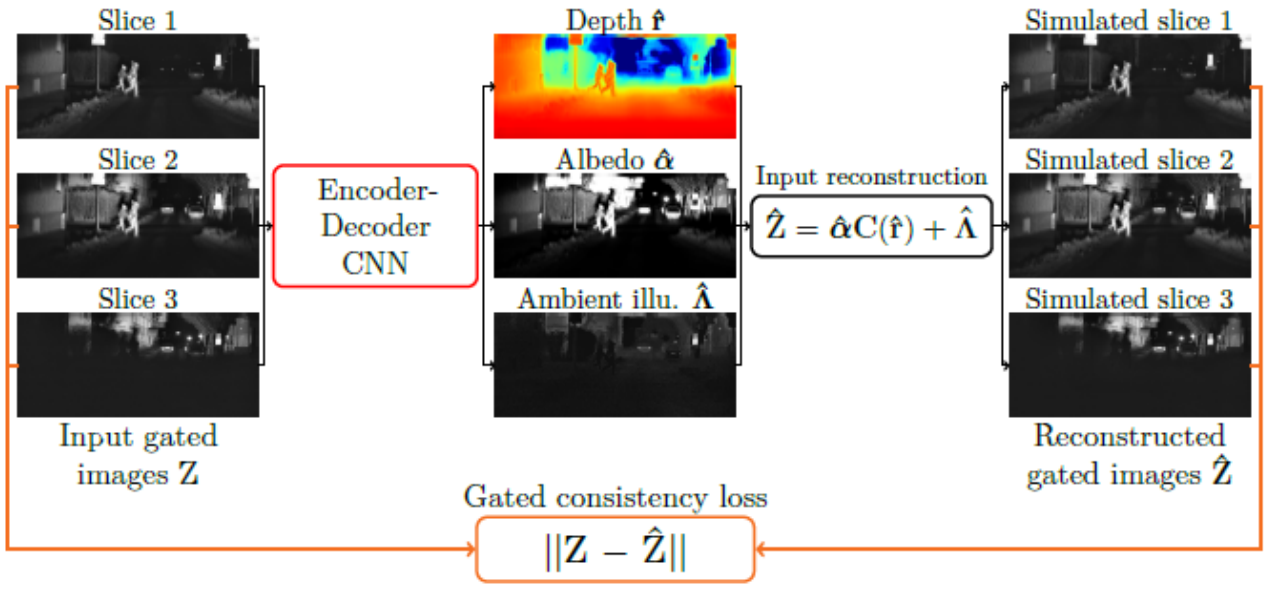

Figure 6.7: Basic principle of a self-supervised gated depth estimation approach. By simultaneously predicting depth, albedo, and ambient illumination, the input gated image can be reconstructed and the absolute error with respect to the real gated images is measured to train the network self-supervised.

A first version of the self-supervised gated depth estimation approach has already been implemented. Currently, this network is being optimized. First results of this approach will be presented in the next deliverable D4.2.

7 Robust 3D Detector for Adverse Weather based on a Gated Camera

7.1 Reference Methods

Today's state-of-the-art methods for 3D object detection are based on LiDAR, stereo, or monocular cameras. LiDAR-based methods achieve the best accuracy, but have a large footprint, high cost, and mechanically-limited angular sampling rates, resulting in low spatial resolution at long ranges. For instance, Figure 7.1 shows examples of a state-of-the art LiDAR-based 3D object detection method [32]. Due to low spatial resolution at long distances, only a few LiDAR points are captured, which makes the detection of objects difficult at those ranges.

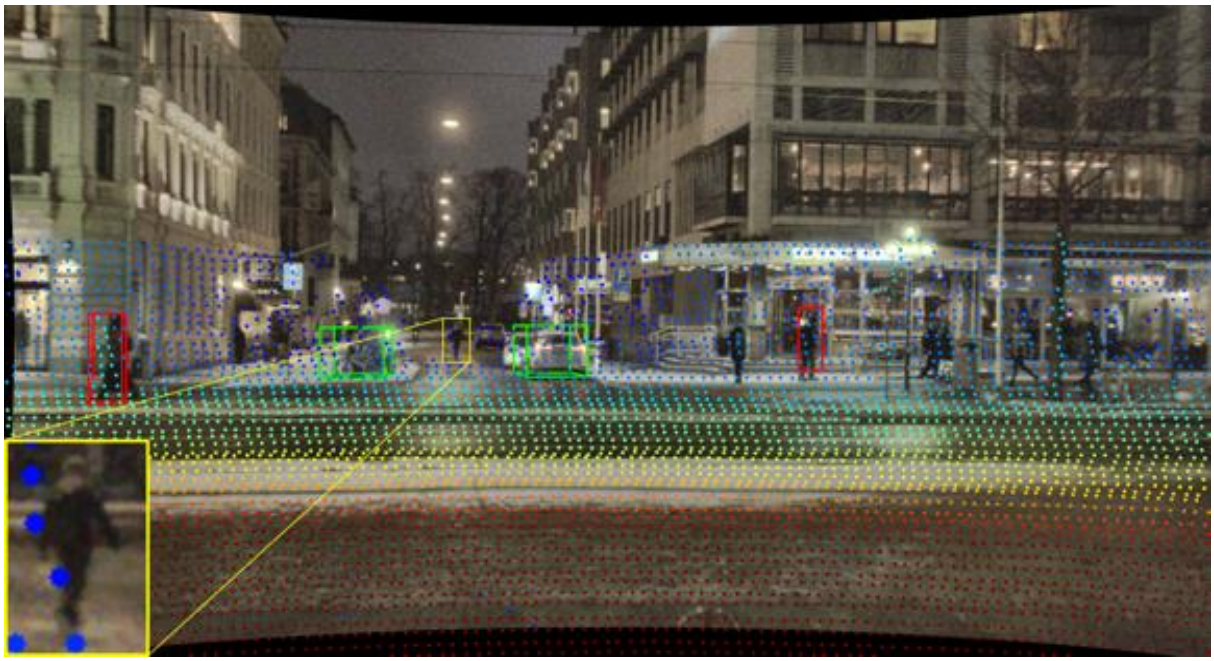

Figure 7.1: LiDAR-based object detection methods struggle at long distance ranges due to low LiDAR resolution at those ranges.

Recent approaches based on low-cost monocular or stereo cameras promise to overcome LiDAR-based methods limitations but struggle in low-light or low-contrast regions as they rely on passive CMOS sensors. Figure 7.2 shows some examples of RGB images in low light conditions.

Figure 7.2: Cameras based on passive CMOS sensors struggle to generate high contrast images in low illumination conditions.

7.2 Architecture of our Gated3D Approach

Within the framework of AI-SEE, we have developed a novel approach for detecting 3D objects from temporal illumination cues in gated images.

Given three gated images, the proposed network determines the 3D location, dimensions, orientation, and class of the objects in the scene. Figure 7.3 shows an example of three input gated slices, a color-coded image by concatenating the gated slices (red being the close-range slice 1), the birds-eye-view of the detection output and a synchronized RGB image.

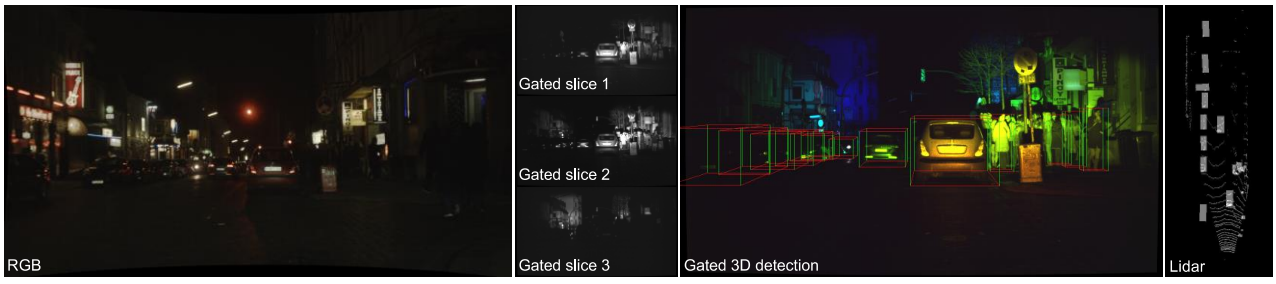

Figure 7.3: Gated3D for 3D object detection from gated images.

The proposed architecture is illustrated in Figure 7.4. Our model is composed of a 2D detection network [33], and a 3D detection network designed to effectively integrate semantic, contextual, and depth information from gated images. The model is trained end-to-end using only 3D bounding box annotations with no additional depth supervision.

In this architecture, the 2D detector predicts bounding boxes that guide the feature extraction using a ResNet[3] backbone. These boxes are also used to estimate frustum segments that constrain the 3D location

[3] ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks.

prediction. In addition to these geometric estimates, the 3D detection network receives the cropped and resized regions of interest extracted from both the input gated slices and the backbone features. To extract contextual, semantic and depth information from the temporal intensity variations of the gated images, our 3D detection network applies two separate convolution streams: one for the backbone features and another for the gated input slices. The resulting features are fed into a sequence of fully-connected layers that predict the 3D location, dimensions, and orientation of the objects.

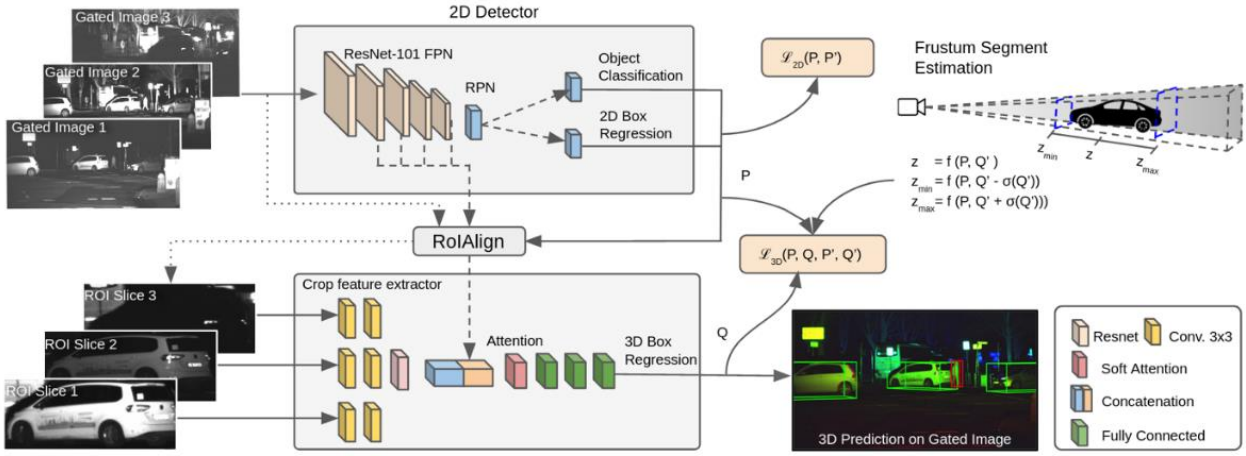

Figure 7.4: Gated 3D architecture. From three gated slices, the proposed Gated3D architecture detects objects and predicts their 3D location, dimension and orientation. Our network employs a 2D detection network to detect ROIs. The resulting 2D boxes are used to crop regions from both the backbone network and input gated slices. Our 3D network estimates the 3D object parameters using a frustum segment computed from the 2D boxes and 3D statistics of the training data. The network processes the gated slices separately, then fuses the resulting features with the backbone features and estimates the 3D bounding box parameters.

Our 3D prediction network fuses the extracted features from both the input gated slices and the backbone features. The gated stream extracts depth cues from the cropped gated input slices with a sequence of convolutions per slice, without parameter sharing. These convolutions consist of three layers with 3 × 3 × 16, 3 × 3 × 32 and 3 × 3 × 32 kernels. The network fuses the three gated features and the backbone features by concatenating along the channel dimension and processing with 5 residual layers. Instead of pooling or flattening the resulting features, an attention sub-network produces softmax attention maps for each feature channel which are used for a weighted sum over the height and width of the features. The resulting feature vectors are fed into two fully connected layers, followed by a final layer that generates eight 3D bounding box coefficients.

We denote an object's predicted 2D bounding box as $P = (c, u, v, h_u, w_v)$, where c is object's class, (u, v) is the bounding box center, and (h_u, w_v) define its height and width, respectively. The 3D detection network takes P and estimates a set of parameters Q, that define a 3D bounding box whose projection is given by P. The problem of estimating Q is ill-posed as given a specific 2D bounding box P, there are an infinite number of 3D boxes that can be projected to P. However, we can restrict the range of locations of Q to a segment of the 3D viewing frustum extracted from P, using the object's approximate dimensions and P. See Figure 7.5 for an illustration.

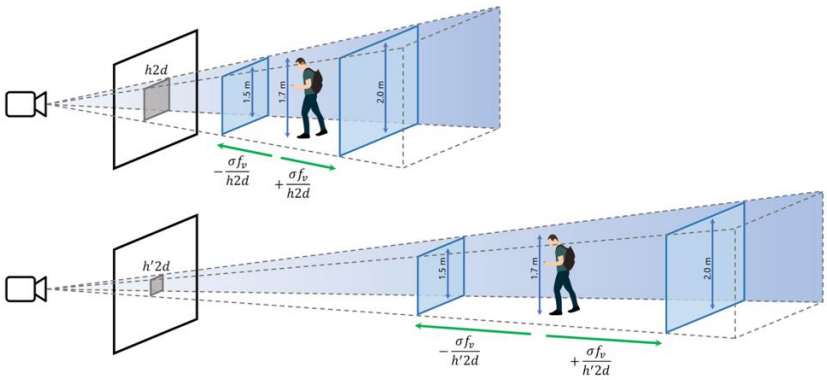

Figure 7.5: There is an infinite number of 3D cuboids that can project to a given bounding box P. However, the object lo- cation can be reasonably estimated using the object height, its projected height, and the vertical focal length.

For each 2D bounding box $P = (c, u, v, w\_u, h\_v)$ generated by the 2D detection network, our 3D bounding box network is trained to estimate the parameters $Q' = (\delta u', \delta v', \delta z', \delta h', \delta w', \delta l', \theta')$, which encode the location (x, y, z), dimensions (h, w, l), and orientation (θ') of a 3D bounding box as discussed in the following.

7.3 3D Location

We estimate the objects location (x, y, z) using its projection over the image space, as well as a frustum segment. Specifically, we define the target $\delta u', \delta v'$ values as

$$\delta u' = (Proj2d_u(x, y, z) - u)/w_u$$
$$\delta v' = (Proj2d_v(x, y, z) - v)/h_v$$

Where $\mathrm{Proj2d_u(x, y, z)}$, $\mathrm{Proj2d_v(x, y, z)}$ represent the coordinates of the 2D projection of (x, y, z) over the image space.

To define the target z, we first define a frustum segment used as a reference for depth estimation. Given an object with height h, we can estimate the object distance to the camera with focal length f_v as

$$f(h_v, h) = \frac{h}{h_v} f_v$$

If we assume that h follows a Gaussian distribution with mean μ_h and standard deviation σ_h, given $P = (c, u, v, w_u, h_v)$ and f_v, we can constrain the distance from the object to the camera to a range of $[f(hv, \mu h - \sigma h), f(hv, \mu h + \sigma h)]$, or, more generally, we deduct that the frustum segment [32] has a length d

$$d = f(h_v, \mu_h + k * \sigma_h) - f(h_v, \mu_h - k * \sigma_h)$$

where k is a scalar that adjusts the segment extent and is inversely proportional to our prediction confidence.

Following these observations, the z coordinate of the 3D bounding box, δz', is given as

$$\delta z' = \frac{z - f(h_v, h)}{d}$$

Note that learning $\delta z'$ instead of the absolute depth z has the advantage that the target value includes a good depth estimation as prior and it is normalized by d, which varies according to the distance from the object to camera. We have found this normalization is the key to estimate the absolute depth of the objects. Intuitively, for higher distances there is a greater localization uncertainty in the labels and as such, the training loss needs to account for this proportionally. Analogous to 2D detectors, this frustum segment can also be considered as an anchor, except its position and dimensions are not fixed, instead using the camera model and object statistics to adjust accordingly.

During training, we use h from ground-truth and during inference, we use the network prediction.

7.4 3D Box Dimensions and Orientation

The target 3D box dimensions are estimated using $\delta h', \delta w', \delta l'$, which are defined as the offset between the mean of the objects dimensions, per class, and the true dimensions:

$$\delta p' = \frac{p - \mu_p}{\mu_p}, \forall p \in \{h, w, l\}$$

To learn the target orientation (observation angle) θ', the orientation is encoded as $(sin_{\theta'}, cos_{\theta'})$, and the network is trained to estimate each parameter separately.

7.5 Loss Functions

Given a 3D box parameters prediction $Q = (\delta u, \delta v, \delta z, \delta h, \delta w, \delta l, sin_\theta, cos_\theta)$, and its corresponding ground-truth box $Q' = (\delta u', \delta v', \delta z', \delta h', \delta w', \delta l', \theta')$, we define our overall loss $\mathcal{L}_{3D}(Q, Q')$ as

$$\mathcal{L}_{3D}(Q, Q') = \alpha * \sum_{l \in \{u,v,z\}} L_{loc}(\delta l - \delta l') + \sum_{d \in \{h,w,l\}} L_{dim}(\delta d - \delta d') + \beta * L_{ori}(sin_{\theta'}, cos_{\theta'}, \theta')$$

where L_{loc} is the location loss, L_{dim} is the dimensions loss, and $L_{ori}(\theta, \theta')$ is the orientation loss. We use α and β to weigh the location and orientation loss, and define these values during training. We define L_{loc} and L_{dim} as $Smooth - L1$, and $L_{ori}(sin_\theta, cos_\theta, \theta')$ as

$$L_{ori}(sin_\theta, cos_\theta, \theta') = (sin_\theta - sin(\theta'))^2 + (cos_\theta - cos(\theta'))^2$$

The method runs at approximately 10 FPS on an Nvidia RTX 2080 GPU in TensorFlow without implementation optimization such as TensorRT.

7.6 Results

We validate the proposed method on real-world driving data acquired with a prototype system in challenging automotive scenarios [8]. Table 7.1 shows *Car* and *Pedestrian* AP for 2D, 3D and BEV detection on the test set. These results demonstrate the utility of gated imaging for 3D object detection. Consistent with prior work [34] both the monocular and stereo baselines show a drop in performance with increasing

distance. Monocular and stereo depth cues for a small automotive baseline of 10-30cm are challenging to find with increasing range.

The proposed Gated3D method offers a new image modality between monocular, stereo and LiDAR measurements. The results demonstrate an improvement over intensity-only methods, especially for pedestrians and at night. Gated3D excels at detecting objects at long distances or in low-visibility situations. Note that pseudo-LiDAR and stereo methods can be readily combined with the proposed method — a gated stereo pair may capture stereo cues orthogonal to the gated cues exploited by the proposed method.

Figure 7.6 shows qualitative examples of our proposed method and state-of-the-art methods. The color-coded gated images illustrate the semantic and space information of the gated data (red tones for closer objects and blue for farther away ones). Our method accurately detects objects even in difficult visibility conditions at both short and long distances. It significantly outperforms other state-of-the-art methods, such as those mentioned in the introduction, which have difficulties, especially in safety-critical applications such as detecting pedestrians at night or in adverse weather conditions.

Table 7.1: Object detection performance over Gated3D dataset. Our method outperforms monocular, stereo and Pseudo-LiDAR methods (bottom part of the table) over most of the short (0-30m), middle (30-50m) and long (50-80m) distance ranges.

(a) Average Precision on *Car* class.

Method	Modality	Daytime Images									Nighttime Images								
		2D object detection			3D object detection			BEV detection			2D object detection			3D object detection			BEV detection		
		0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m
PointPillars [32]	Lidar	90.12	82.83	56.63	91.51	84.63	54.28	91.59	86.54	54.71	90.73	84.88	54.22	90.29	87.40	52.32	90.29	87.51	52.60
M3D-RPN [5]	RGB	90.44	89.29	62.76	53.21	13.26	10.52	**60.80**	16.16	10.52	90.85	80.64	59.76	51.18	20.76	2.73	52.53	21.39	2.74
Stereo-RCNN [35]	Stereo	81.56	81.07	78.08	**54.17**	17.16	6.17	57.92	17.69	6.26	81.73	81.03	70.85	47.36	17.21	13.02	**53.81**	18.34	13.08
Pseudo-Lidar	Gated	81.74	81.33	80.88	26.17	16.06	10.27	26.94	17.26	10.87	89.35	89.02	88.31	36.58	23.05	19.88	39.50	28.68	22.82
Pseudo-Lidar++ [64]	Gated	81.74	80.29	81.59	30.44	15.47	11.76	32.49	16.97	12.83	90.21	81.75	81.78	36.36	21.93	**22.39**	37.46	23.12	**23.63**
PatchNet [41]	Gated	90.46	81.74	89.78	23.91	10.86	7.34	24.87	11.33	7.84	**90.87**	**89.86**	88.89	23.74	16.79	7.16	25.15	17.76	8.29
Gated3D	Gated	**90.78**	**90.55**	**90.91**	52.15	**28.31**	**14.85**	52.31	**29.26**	**15.02**	90.84	81.82	**90.33**	**51.42**	**25.73**	12.97	53.37	**29.13**	13.12

(b) Average Precision on *Pedestrian* class.

Method	Modality	Daytime Images									Nighttime Images								
		2D object detection			3D object detection			BEV detection			2D object detection			3D object detection			BEV detection		
		0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m	0-30 m	30-50 m	50-80 m
PointPillars [32]	Lidar	70.08	49.03	0.00	69.71	45.24	0.00	70.53	48.07	0.00	69.97	43.32	0.00	71.25	41.21	0.00	70.99	43.61	0.00
M3D-RPN [5]	RGB	79.08	66.41	36.98	26.20	14.50	9.84	30.68	17.47	10.07	78.36	62.99	36.76	25.09	6.43	2.07	26.42	7.69	2.74
Stereo-RCNN [35]	Stereo	88.57	75.63	59.82	48.58	**23.26**	7.77	50.11	**25.10**	8.38	80.38	69.13	60.94	46.09	21.63	11.57	47.58	25.47	11.84
Pseudo-Lidar	Gated	77.87	78.38	69.11	6.19	4.59	2.15	10.28	9.14	4.13	80.34	78.61	67.78	7.53	9.58	1.62	14.27	15.72	5.55
Pseudo-Lidar++ [64]	Gated	77.89	77.95	60.88	9.19	2.36	3.30	14.32	5.66	4.10	79.84	79.57	54.42	7.37	7.21	2.06	12.92	11.99	5.64
PatchNet [41]	Gated	**90.48**	80.75	69.56	32.88	18.05	5.62	39.45	20.27	9.77	81.50	**88.62**	65.43	15.37	13.37	6.75	21.60	18.15	8.46
Gated3D	Gated	89.72	**81.47**	**86.73**	**50.94**	20.59	**14.14**	**53.26**	22.15	**16.51**	**81.52**	81.23	**80.18**	**48.53**	**23.99**	**14.98**	**49.82**	**25.57**	**15.46**

Figure 7.6: Qualitative comparison against baseline methods. Bounding boxes from the proposed method are tighter and more accurate than the state-of-the-art methods. This is seen in the second image with the other methods showing large errors in pedestrians.

List of abbreviations

ABBREVIATION	MEANING
ADS	Automatic Driving System
AUPRC	Area Under Precision-Recall Curve
BEV	Bird-Eye-View
CMOS	Complementary Metal Oxide Semiconductor
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FC-Layer	Fully-Connected-layer
FMCW	Frequency-Modulated Continuous-Wave
FPGA	Field-Programmable Gate Array
GDP	Gated Delay Profile
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
MIMO	Multiple Input-Multiple Output
NIR	Near InfraRed
RADAR	Radio Detection and Ranging
RANSAC	RANdom SAmple Consensus
ResNet	Residual Networks
RGB	Red Green Blue
RIP	Range-Intensity-Profile
ROS	Robot Operating System
SNR	Signal-to-noise ratio
ToF	Time-of-Flight

References

[1] O. David and S. Inbar, "Laser gated camera imaging system and method," 2008.

[2] Z. Zhang, "A flexible new technique for camera calibration," *#TPAMI#,* vol. 22, 2000.

[3] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics,* vol. 11, pp. 431-441, 1963.

[4] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics,* vol. 2, pp. 164-168, 1944.

[5] Administration, U.S. Department of Transportation: Federal Highway, "How Do Weather Events Impact Roads?," [Online]. Available: https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm.

[6] European Commission Directorate-General for Mobility and Transport, "Road safety in the European Union : trends, statistics and main challenges," Publications Office, 2018.

[7] European Commision, "Annual Accident Report," 2018.

[8] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer and F. Heide, "Seeing through fog without seeing fog: deep multimodal fusion in unseen adverse weather.," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[9] H. Koschmieder, "Therie der Horizontalen Sichtweite," *Beiträge zur Physik der freien Atmosphäre,* 1924.

[10] C. Sakaridis, D. Dai and L. Van Gool, "Semantic Foggy Scene Understanding with Synthetic Data," *International Journal of Computer Vision,* p. 973–992, 2018.

[11] J. Chen and R. Chang, "Pyramid Stereo Matching Network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[12] X. Cheng, Y. Zhong, Harandi and M. Harandi, "Hierarchical Neural Architecture Search for Deep Stereo Matching," in *https://arxiv.org/abs/2010.13501*, 2020.

[13] S. Zheng, E. Tseng, M. Bijelic, W. Ritter and F. Heide, "ZeroScatter: Domain Transfer for Long Distance Imaging and Vision through Scattering Media," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] H. Cesar, V. Bankiti, A. Lang and et. al, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite.," in *IEEE Conference on Computer Vision and Pattern (CVPR)*, 2012.

[16] M. Hahner, C. Sakaridis, D. Dai and L. Van Gool, "Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[17] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, Y. Fisher, D. Dai, L. Van Gool and M. Hahner, "LiDAR Snowfall Simulation for Robust 3D Object Detection," in *https://arxiv.org/abs/2203.15118*, 2022.

[18] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.," 1980.

[19] M. Tan, V. Quoc and J. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.

[20] M. R. Ibrahim, J. Haworth and T. Cheng, "WeatherNet: Recognising Weather and Visual Conditions from Street-Level Images Using Deep Residual Learning," in *MDPI*, 2019.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks.," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization.," in *International Conference on Learning Representations.*, 2014.

[23] A. Foi, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing,* pp. 2609-2629, 12 2009.

[24] J. Busck and H. Heiselberg, "Gated viewing and high-accuracy three-dimensional laser radar," *Applied optics,* vol. 43, no. 24, p. 4705–4710, 2004.

[25] J. Busck, "Underwater 3-D optical imaging with a gated viewing laser radar," *Optical Engineering,* vol. 44, no. 11, p. 116001, 2005.

[26] M. Laurenzis, F. Christnacher and D. Monnin, "Long-range three-dimensional active imaging with superresolution depth mapping," *Optics Letters,* vol. 32, no. 21, p. 3146–3148, 2007.

[27] M. Laurenzis, F. Christnacher, N. Metzger, E. Bacher and I. Zielenski, "Three-dimensional range-gated imaging at infrared wavelengths with super-resolution depth mapping," SPIE, 2009, p. 1166–1171.

[28] C. Jin, X. Sun, Y. Zhao, Y. Zhang and L. Liu, "Gain-modulated three-dimensional active imaging with depth-independent depth accuracy," *Optics Letters,* vol. 34, no. 22, p. 3550–3552, 2009.

[29] Z. Xiuda, Y. Huimin and J. Yanbing, "Pulse-shape-free method for long-range three-dimensional active imaging with high linear accuracy," *Optics Letters,* vol. 33, no. 11, p. 1219–1221, 2008.

[30] T. Gruber, M. Bijelic, F. Julca-Aguilar and F. Heide, "Gated2depth: Real-time dense lidar from gated images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[31] T. Gruber, M. Kokhova, W. Ritter, N. Haala and K. Dictmayer, "Learning Super-resolved Depth from Active Gated Imaging," IEEE, 2018.

[32] C. Qi, W. Liu, C. Wu, H. Su and L. Guibas, "Frustum pointnets for 3d object detection from RGB-D data.," in *IEEE Conference on Computer Vision and Pattern Recognition,*, 2018.

[33] K. He, P. kioxsari, P. Dollar and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[34] P. Li, X. Chen and S. Shen, "Stereo R-CNN based 3d object detection for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.