Deliverable D4.2

Sensor Enhancement and Enrichment

Dissemination level	**PU**
Version	**1.0**
Lead contractor	**ALG**
Due date	**30.09.2022**
Version date	**27.09.2022**

Document information

Authors

Dr. Werner Ritter – Mercedes Benz AG

Mario Bijelic – Mercedes Benz AG

Dominik Scheuble – Mercedes Benz AG

Stefanie Walz – Mercedes Benz AG

Andrea Ramazzina – Mercedes Benz AG

Matthias Schulze – Algolux (Germany) GmbH

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

Table of contents

List of figures

List of tables

1 Executive Summary

This is the second deliverable in the work package *WP4 Sensor Fusion and AI*. It is dedicated to the enhancement and enrichment of raw sensor data streams degraded by adverse weather using novel generative neural network (NN) models.

In this deliverable, we first describe a signal enhancement method to improve detection results for LiDAR systems in the adverse weather case snow. The basis for this improvement was a sufficiently large sample of learning data, which we were able to provide in this case via a simulation method we developed in the project, which we describe in detail in this deliverable.

Next, as an additional example of a signal enhancement method, we present a defogger for image data developed in AI-SEE.

Finally, examples of combined signal enhancement with signal enhancement using two depth estimation methods are given in this document: Depth estimation using a newly developed wide-base stereo approach and depth estimation using depth slices from a monocular gated camera, method to significantly improve detection results.

2 Introduction

The second deliverable in work package *WP4 Sensor Fusion and AI* is to the enhancement and enrichment of raw sensor data streams degraded by adverse weather using novel generative neural network (NN) models.

In this deliverable, Chapter 3 first describes a signal enhancement method that significantly improves the detection results of LiDAR systems in snow. The basis and main ingredient for this improvement was a sufficiently large sample of learning data, which in this case we were able to provide via a simulation method we developed in the project. This augmentation-based simulation method is described in detail in this chapter.

In chapter 4 we describe with defogging method for camera images developed in AI-SEE, an image enhancement method developed in AI-SEE. The NN of this defogging method was also trained for the most part based on artificially generated adverse weather data and verified using real world adverse weather data.

In chapter 5, we address combined enhancement and enrichment methods for extracting depth information from camera data: First, an approach to extract depth information from a "standard" wide-base stereo camera focused on the detection of small objects at large distances (lost-cargo detection up to 150 m distance) and second, an approach to extract depth information from the depth section images of a monocular gated camera.

3 Detection enhancement of LiDAR in snow based on simulated snowy LiDAR data

It is essential for the development and verification of NN methods to have a sufficiently large amount of suitable learning and test data. Since this data cannot be obtained on the required scale via measurement campaigns in the real world, the only way out is to generate this data artificially, i.e. to simulate it.

The quality of the enhancement methods developed on the basis of this artificially generated adverse weather data stands and falls with the quality of the simulation. The better the simulated world matches the real world (from the point of view of a perception system), the smaller will be the difference of the behaviour of a perception system during the transition between the virtual and the real world. The goal here is that this difference is no longer noticed by the perception system.

The great advantage of performing the simulation together with the development of perception system is that the simulation can be directly examined by the only evaluation system relevant to this simulation, namely the perception system of an ADS itself. If the perception system of the ADS behaves in the world of simulated data as in the world of real data (e.g. same detection quality measures with a sufficiently large sample), then the simulation is perfect and the so-called domain gap is zero.

As part of AI-SEE, we have developed (in close coordination with the publicly funded BMWi project VVM[1]) a simulation of snowfall for LiDAR sensor data. This snowfall sensor data is simulated by augmenting snowfall on the sensor data acquired in "normal" weather. The novel method we developed in AI-SEE is presented in the following sub-chapters.

3.1 Captured Ground Truth Dataset

Our experiments are carried out on the STF dataset [1]. It provides 12997 annotated samples with accurate 3D bounding boxes for object detection of cars, pedestrians, and cyclists in various weather conditions including light fog, dense fog and snow. Without denying the importance of the pedestrian and cyclist classes, in the main paper, we focus on the most dominant class, i.e.\ cars.

In total, 3469 frames in clear conditions can be used for training and 3916 frames in snowy conditions are provided. To allow a more fine grained evaluation we split the 3916 samples in the snow test set based on the intensity of the snowfall into two different subsets, termed light snowfall and heavy snowfall, with 2512 and 1404 samples respectively. Inspired by the CADC dataset [2] we perform this split by leveraging the DROR algorithm [3]. The light snowfall split contains frames where DROR [3] would filter 10-79 points from a 10x2x2m box in front of the ego vehicle, while the heavy snowfall contains at least 80 of such points within this box.

[1] VVMethods / Verification and validation methods of automated vehicles Level 4 and 5: The focus in VVM is on verification models, and the focus in AI-SEE is on increasing the all-weather capability of perceptual systems. Since these focuses are very complementary and cannot be done in isolation, we are doing the work together and thus generate significant synergy effects.

As parametrization, we set the horizontal angular resolution to 0.45° as we noticed that the original parametrization of 0.35° is too aggressive and removes too many uncluttered points.

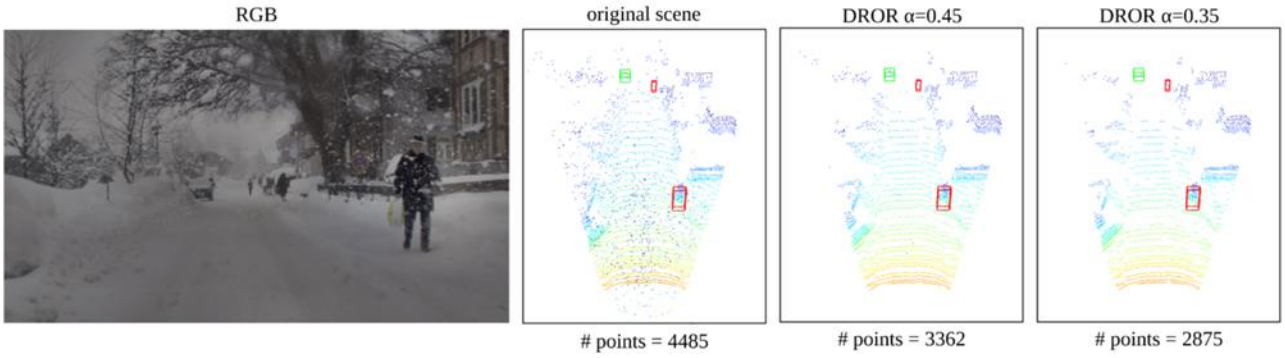

Figure 3.1: The figure shows the camera image of a scene from the heavy snowfall test set (the first column) and the original point cloud including our proposed box (in blue) for classifying the snowy test set in two subsets (the second column). The third column shows the original DROR [3] parametrization proposed by its authors and the fourth column shows our adjusted parametrization. Best viewed on a screen and zoomed in.

This parametrization generally removes just as much snowfall clutter but keeps more valid points remaining in the scene as shown in Figure 3.1. To enforce as many as possible snowy samples in our test set we discard all samples with less than 9 cluttered points.

This way we make sure that less clear samples make it to the corrected snowy test set. In the end, we exclude 552 frames from the initial "snowy" test set and are left with 3916 frames which contain at least 10 points in our proposed box in front of the vehicle classified as clutter by the DROR [3] filter.

2512 frames, where DROR classifies 10-79 points from within the aforementioned box as clutter, construct the **light snowfall** test set and the remaining 1404 frames where DROR classifies at least 80 (and in the most extreme scene up to 713) points as snowfall clutter construct the **heavy snowfall** test set. Figure 3.2 shows an example with 10, 80, and the aforementioned, most extreme scene with 713 points counted by DROR [3] as clutter within our box, respectively.

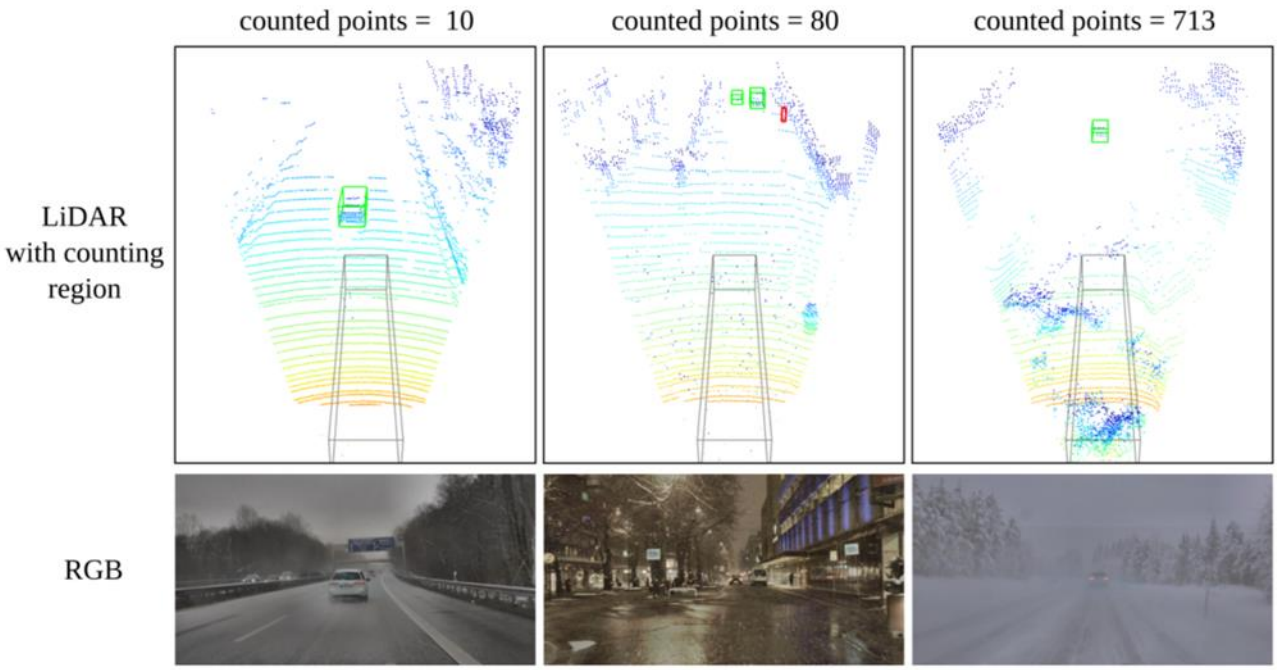

Figure 3.2: Example scenes with the thresholds used for creating the light and heavy snowfall test splits.

The estimated number of scattered particles per scenes yields an exponential distribution and is shown in Figure 3.3. As extreme snowfall cases are rare in real world captures, this shows the importance of creating data augmentation for heavy adverse snowfall.

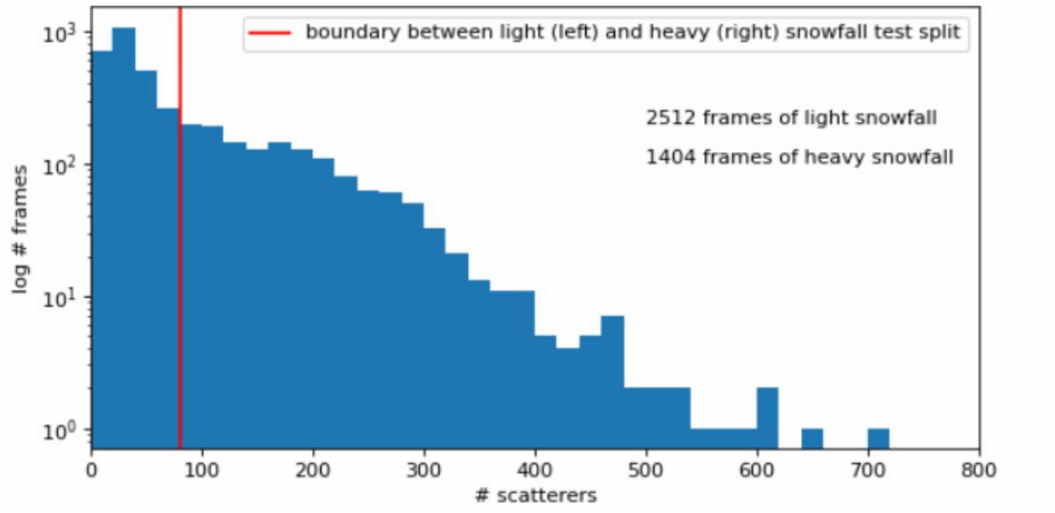

Figure 3.3: Histogram of scatterers per frame (with bin size 20).

3.2 Snow Simulation Method

The method aims to reproduce disturbances introduces by adverse weather conditions in snowy conditions. Therefore, we built a physical model accurately modelling the interaction of laser beams with

individual snowflakes. Those simulations are subsequently used to create realistically looking snowy point clouds from clear easy to collect samples. Then, the partially synthetic point clouds with our snowfall are used as training data for optimizing state-of-the-art 3D object detection methods, so that the learned models are more robust under snowfall. The hope is that our physically based simulation is realistic enough to relieve us from the need for real snowy training samples. We benchmark the models trained in this regime on the challenging real snowy subset of the STF dataset [1] and find that the models trained on our simulated snow consistently achieve significant performance gains over baseline models trained only on clear weather and competing simulation methods.

Next we begin with the derivation of the pulse propagation of individual snowflakes.

Pulse propagation in free space can be modelled with geometrical optics for cost-effective LiDAR systems. Such systems apply an array of synchronized near-field infrared pulse emitters Tx and avalanche photodiodes (APDs) as receivers Rx depicted in Figure 3.4 and described in [4].

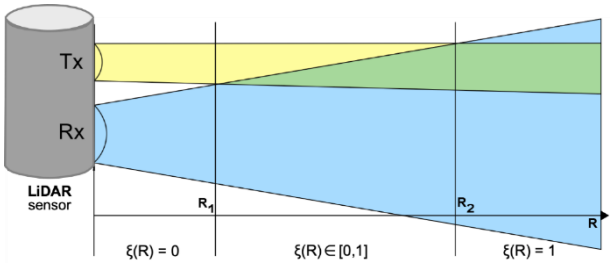

Figure 3.4: Sketch of a LiDAR sensor where the transmitter Tx and the receiver Rx do not have coaxial optics, but have parallel axes (called a bistatic beam configuration).

The sent-out laser pulse P_0 is reflected by a solid scene object, often referred to as target, with reflectivity ρ_0, and it is captured by the receiver, providing the time delay τ of the captured echo and its corresponding power P_R. The object distance R is calculated by applying $R = c\tau$, where c is the speed of light. The 3D position $[x, y, z]$ of the object is obtained by using the direction in which the pulse was emitted. For extended objects, geometric optics [5] can be applied to model the received power $P_R(R)$ following

$$P_R(R) = C_A \, P_0 \, \rho_0 \, \frac{\cos(\alpha_{in})}{R^2} \, ,$$

which holds for objects with a diameter larger than the beam diameter at distance R and requires additional information about (i) the incident angle α_{in} and (ii) the system constant C_A independent of range and time.

However, the received laser power is typically corrected [5], as C_A differs for each scanning layer due to different optics and beam divergences. Four different levels of intensity calibration can be reported according to [6].

For the Velodyne HDL-S3D sensor used in our experiments, a beam divergence correction is applied following the sensor manual [7]. This correction is defined as

$$i = P_R + f_s \times \left(f_o - \left(1 - \frac{R}{R_{max}} \right) \right)^2 ,$$

where f_s is the focal slope and f_o is the focal offset.

The parameters for each laser are retrieved from the factory side calibration. Before applying the proposed simulation methods, we first retrieve the raw intensities by inverting this intensity calibration.

In snowfall, the optical medium contains particles which are smaller than the beam diameter, so Mie scattering and the exact spatial distribution of the particles must be taken into account [8].

Pulse propagation in the presence of scattering particles is described by a linear model introduced in [8], which is valid for non-elastic scattering. This model expresses the range-dependent received power P_R as a time-wise convolution between the time-dependent transmitted signal power P_T and the impulse response H of the optical system:

$$P_R(R) = C_A \int_0^{\frac{2R}{c}} P_T H\left(R - \frac{ct}{2}\right) dt,$$

with the time signature of the transmitted pulse given by

$$P_T(t) = \begin{cases} P_0 \sin^2\left(\frac{\pi}{2\tau_H} t\right), & for \quad 0 \le t \le 2\tau_H \\ 0, & otherwise \end{cases}$$

τ_H is the half-power pulse width, set to 10 ns for the Velodyne HDL-S3D sensor. The impulse response H can be factored into the impulses responses of the optical channel, H_C, and the target, H_T:

$$H(R) = H_C(R)\ H_T(R).$$

H_C depends on the beam divergence, the overlap of transmitter and receiver described by $\xi(R)$ as well as the transmittance $T(R)$ of the medium through

$$H_C(R) = \frac{T^2(R)}{R^2}\ \xi(R).$$

The transmittance $T(R)$ is equal to 1 in the part of the medium that is not occupied by snow particles, assuming absence of other scattering elements. The overlap $\xi(R)$ can be geometrically derived from Figure 3.4 as

$$\xi = \begin{cases} 0, & R \le R_1 \\ \frac{R - R_1}{R_2 - R_1}, & R_1 < R < R_2 \\ 1, & R_2 \le R \end{cases}$$

The impulse response of the target, H_T, allows us to model snow particles as we detail in the following.

Scene reflection defines the particle interaction with the laser pulse through H_T. For an extended solid target object we can write

$$H_T(R) = \rho_0 \delta(R - R_0),$$

with ρ_0 being the reflectivity of the object and δ the Dirac delta function. However, in snowfall, apart from the solid target object, the laser beam is also partially reflected by snow particles.

We model snow particle j as a spherical object with reflectivity ρ_s, diameter D_j following the distribution introduced in [9] and distance R_j from the sensor, placed uniformly at random around the sensor so that it does not intersect with any other particle. The number of particles is chosen according to the snowfall rate,

typically ranging in 0-2.5 mm/h. Particles can occlude each other and the target object, as illustrated in Figure 3.5.

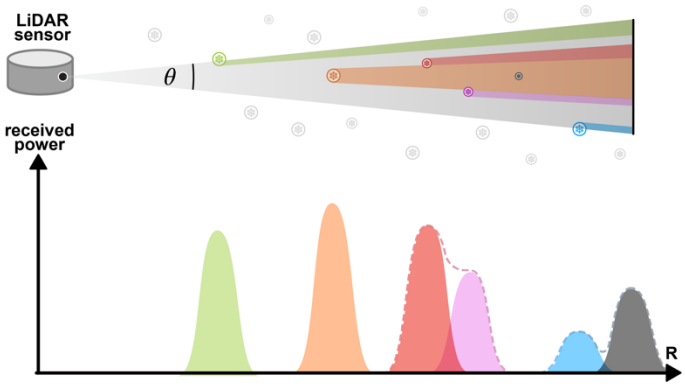

Figure 3.5: Snow particles interfering a single LiDAR beam (top). Schematic plot of corresponding received power echoes (bottom). Note how the received power of individual targets can overlap with each other ($c\,\tau_H \approx 3\,m$ with $\tau_H = 10\,ns$)

Thus, each particle j reflects only a fraction $\frac{\theta_j}{\Theta}$ of the opening angle Θ of the beam, also letting a fraction $\frac{\theta_0}{\Theta}$ of the beam reach the target.

Assuming $D_j \ll c\tau_H$ for all j, we can write

$$H_T(R) = \frac{1}{\Theta}\left(\rho_0\,\theta_0\,\delta(R-R_0) + \rho_s\sum_{j=1}^{n}\theta_j\,\delta(R-R_j)\right),$$

with $\Theta = \theta_0 + \sum_{j=1}^{n}\theta_j$.

Plugging previous equations into one another, the received power in snowfall is

$$P_{R,snow}(R) = P^0_{R,snow}(R) + \sum_{j=1}^{n} P^j_{R,snow}(R)\,,$$

where

$$P^j_{R,snow} = \frac{C_A P_0 \rho_s \theta_j \xi(R_j)}{\Theta\,\mathrm{R_j^2}}\int_0^{2\tau_H}\sin^2\left(\frac{\pi}{2\tau_H}t\right)\delta\left(R-\frac{ct}{2}-R_0\right)dt,$$

$$\Longrightarrow P^j_{R,snow} = \begin{cases}\dfrac{C_A P_0 \rho_s \theta_j \xi(R_j)}{\Theta\,\mathrm{R_j^2}}\sin^2\left(\dfrac{\pi(R-R_j)}{2\tau_H}\right), & for\ R_j \le R \le R_j + c\tau_H \\ 0\,, & otherwise\end{cases}$$

$P^0_{R,snow}(R)$ can be derived by substituting $(\theta_j\ R_j\ \rho_s\ )$ with $(\theta_0\ R_0\ \rho_0)$ on the right-hand side of the previous equation.

The received power is thus a superposition of multiple echoes, each associated with an object (snow particle or target object), as depicted in Figure 3.5. Crucially, the magnitude of each echo depends on the

angle θ_j and the inverse square of the distance R_j of the respective object from the sensor. In this work, we retrieve the maximum peak of the received power as the LiDAR return.

Thus, if a peak owing to snow particles is higher than the peak associated to the target object, the true echo is missed and a cluttered point is added to the simulated point cloud at the range of the former peak.

Otherwise, the target object intensity is attenuated according to its occlusion percentage. Our complete snowfall simulation is presented in Figure 3.7. In Figure 3.6 we show a wintry example scene, once augmented with our snowfall simulation and once with the one proposed in LISA [9] as comparison.

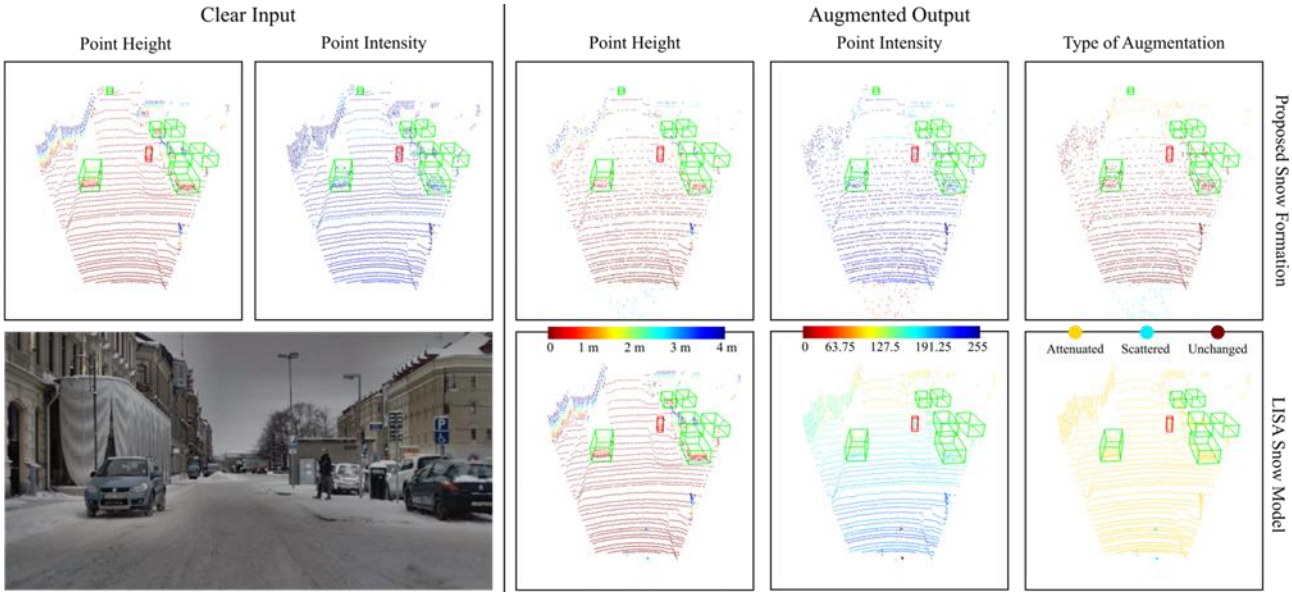

Figure 3.6: Simulated snowfall corresponding to a snowfall rate of $r_s = 2.5\ mm/h$. The left block shows the clear undisturbed input. The right block shows our snowfall simulation (top) and the snowfall simulation in LISA [10] (bottom). Note that we simulate the scattering realistically and only attenuate points which are affected by individual snowflakes instead of attenuating all points based on their distance.

Algorithm 1 LiDAR snowfall simulation

procedure SNOWFALL($\mathbf{pc}, n_l, \tau_H, R_{\text{max}}, \Theta, r_s, \rho_s, \rho_0$)
  for l in n_l **do** ▷ for each layer l
    $\mathbf{pc}_l \leftarrow \mathbf{pc}$.SELECT(layer $= l$)
    $f_d, f_s, i_{\text{max}} \leftarrow$ LOAD_CALIB(l)
    $f_o \leftarrow (\frac{1-f_d}{13100})^2$ ▷ focal offset [50]
    $\mathbf{s} \leftarrow$ SAMPLE_SNOWFLAKES(R_{max}, r_s) ▷ in 2D [15]
    for $\mathbf{p}$ in $\mathbf{pc}_l$ **do** ▷ for each point in layer l
      $x, y, z, i \leftarrow \mathbf{p}$
      $R_0 \leftarrow \|\mathbf{p}\|_2$
      $\mathbf{t} \leftarrow$ GET_PARTICLES_IN_BEAM($\mathbf{s}, x, y, R_0, \Theta$) ▷ in 2D
      if len($\mathbf{t}$) > 1 **then** ▷ otherwise no interference
        $\mathbf{P}_{R,\text{snow}} \leftarrow \mathbf{0}$ ▷ initialize with zeros
        for i, R, θ in $\mathbf{t}$ **do** ▷ for each target
          if $R = R_0$ **then** ▷ original target
            $P_R \leftarrow i - f_s \left| f_o - \left(1 - \frac{R}{R_{\text{max}}}\right) \right|^2$
            $C_A P_0 \leftarrow \frac{P_R}{\rho_0} {R_0}^2$ ▷ follows from Eq. (1)
          else ▷ snowflake
            $P_R \leftarrow \rho_s \, i_{\text{max}}$
            $C_A P_0 \leftarrow \frac{P_R}{\rho_0}$
          end if
          $\mathbf{P}_{R,\text{snow}} \mathrel{+}=$ Eq.(11)$(C_A P_0, R, \rho, \tau_H, \theta, \Theta)$
        end for
        $P_R \leftarrow \max(\mathbf{P}_{R,\text{snow}})$
        $R^* \leftarrow \arg\max(\mathbf{P}_{R,\text{snow}}) - c\frac{\tau_H}{2}$
        $i \leftarrow P_R + i_{\text{max}} f_s \left| f_o - \left(1 - \frac{R^*}{R_{\text{max}}}\right) \right|^2$
        $(x, y, z) \leftarrow \frac{R^*}{R_0} \times (x, y, z)$
        $\mathbf{p} \leftarrow x, y, z, i$
      end if
    end for
  end for
  return $\mathbf{pc}$
end procedure

Figure 3.7: Summary of the Algorithm adding snow particles to clear real-world LiDAR point clouds.

3.3 Results

For evaluation, we use the 3D object detection metrics defined in the KITTI evaluation framework [10] and in [11]. Specifically, [11] introduces an extension to the KITTI metrics by reporting the results with respect to the object distance. Since the weather effects of snow, detailed in the previous section 3.2, are distance-dependent, we opt for following their extension and report results in the intervals as in [12]. Additionally, we follow [13] and report average precision (AP) at 40 recall positions to provide a fair comparison. Other than that, we use the typical overlap thresholds defined in [13].

To mitigate potential statistical fluctuations, we report for each experiment the average performance over three independent training runs.

In total, we investigate the effectiveness of our snowfall simulation scheme for seven well-known 3D object detection methods [14], [15], [16], [17], [18], [19], [20]. We compare our approach to a clear-weather baseline and two competing adverse weather simulation methods, one for fog [21] and one for snowfall [22]. Additionally, we compare to de-noising the point clouds using DROR [4]. To train the detection models, we use OpenPCDet [23] and follow the default training configurations for each method. All methods are trained from scratch.

We choose to apply our simulation(s) to every 10-th training sample, for which the snowfall rate is sampled from [0, 0.5, ..., 2.5]mm/h, and set the sensor constants $\tau_H = 10\ ns$,$R_{max} = 120\ m$, $\Theta = 0.003\ °$, $\rho_s = 0.9$ and $\rho_0 = \frac{10^{-6}}{\pi}$. The exact same settings are used for [10].

We present the quantitative results in Table 3.1. In reading Table 3.1, the reader should first focus on the columns showing AP across the entire evaluation range of [0-80]m.

Table 3.1 Comparison of simulation methods for 3D object detection in snowfall on STF [1]. We report 3D average precision (AP) of moderate cars on three STF splits: the heavy snowfall test split with 1404 samples, the light snowfall test split with 2512 samples and the clear-weather test split with 1816 samples.

Detection method	Simulation method	heavy snowfall ↑				light snowfall ↑				clear weather ↑			
		0-80m	0-30m	30-50m	50-80m	**0-80m**	0-30m	30-50m	50-80m	**0-80m**	0-30m	30-50m	50-80m
PV-RCNN [39]	*None*	39.69	65.05	36.14	8.03	41.13	69.24	39.72	**11.68**	45.36	**72.34**	42.48	10.53
	Fog [16]	38.19	64.72	33.38	7.49	39.82	68.41	38.68	9.65	43.37	71.05	40.03	9.90
	DROR [6]	38.57	64.27	35.40	8.07	39.33	66.73	38.14	10.51	41.44	67.76	38.48	9.44
	LISA [22]	39.21	64.21	35.34	8.64	41.60	69.15	**41.08**	11.15	45.30	71.06	42.86	11.45
	Ours-snow	41.61	67.44	**37.47**	8.84	41.20	68.79	40.20	11.13	45.61	72.14	**43.40**	11.21
VoxelRCNN-Car [9]	*None*	39.47	65.14	36.29	6.83	41.25	69.12	39.86	**11.81**	45.19	72.33	**43.20**	10.69
	Fog [16]	40.06	65.58	36.78	7.33	41.10	68.93	39.25	10.98	44.46	71.67	41.78	10.84
	DROR [6]	38.16	64.97	33.23	6.83	38.48	66.93	35.68	9.97	40.65	67.94	36.85	8.45
	LISA [22]	39.06	66.61	33.56	6.93	40.68	68.80	38.78	10.75	45.03	72.05	41.96	10.59
	Ours-snow	**41.20**	**68.27**	37.18	**7.90**	**41.75**	**70.22**	**40.95**	11.78	44.52	**72.40**	42.23	10.39
CenterPoint [59]	*None*	38.68	63.50	34.20	8.23	40.91	68.36	39.96	10.98	44.11	71.66	40.97	9.76
	Fog [16]	38.82	64.43	33.70	8.69	40.82	68.71	40.05	10.48	43.79	70.34	41.22	10.77
	DROR [6]	38.42	64.47	34.31	8.27	38.69	65.62	37.59	10.26	40.80	67.62	36.61	8.69
	LISA [22]	38.11	63.74	33.58	7.67	40.26	68.16	39.25	11.20	44.70	71.54	41.46	**11.23**
	Ours-snow	**37.73**	**66.44**	29.54	6.46	38.23	**67.72**	36.02	7.41	**43.41**	**71.00**	**40.63**	**9.74**
Part-A² [42]	*None*	36.59	63.50	30.17	**6.86**	38.03	65.83	35.95	**9.39**	42.81	70.27	39.90	9.18
	Fog [16]	35.98	62.23	30.94	6.39	38.17	66.07	37.12	8.63	41.82	67.44	39.73	9.39
	DROR [6]	35.85	65.36	27.99	6.13	35.43	63.50	32.87	7.95	39.48	66.92	35.18	8.61
	LISA [22]	37.12	65.57	30.05	5.96	38.04	66.62	36.72	8.25	41.92	70.02	38.84	7.79
	Ours-snow	**37.73**	**66.44**	29.54	6.46	38.23	**67.72**	36.02	7.41	**43.41**	**71.00**	**40.63**	**9.74**
PointRCNN [41]	*None*	36.68	61.74	33.25	6.14	39.04	66.90	39.72	9.28	41.79	**68.58**	**40.34**	7.96
	Fog [16]	36.56	62.93	33.03	5.52	38.37	65.71	39.60	8.50	41.28	67.79	38.82	7.88
	DROR [6]	36.14	62.64	31.64	5.28	36.31	63.52	36.62	7.77	39.08	64.96	36.54	7.70
	LISA [22]	36.68	62.85	31.80	5.78	38.08	65.20	38.96	8.96	**41.80**	68.17	39.79	**8.35**
	Ours-snow	**37.59**	**63.99**	33.63	6.15	38.60	65.18	39.20	9.47	41.43	67.61	39.89	8.15
SECOND [57]	*None*	36.08	61.53	30.92	6.60	37.77	**65.68**	36.06	9.80	42.10	67.82	39.52	10.81
	Fog [16]	36.08	61.65	31.25	7.65	37.31	64.27	35.43	**10.55**	42.34	**69.85**	39.20	10.13
	DROR [6]	35.04	60.72	28.79	7.88	35.09	62.24	32.09	8.85	38.96	64.74	35.50	9.76
	LISA [22]	35.90	59.31	**32.81**	7.44	**38.07**	64.38	**38.47**	10.32	41.75	67.01	38.24	**11.47**
	Ours-snow	36.83	61.94	31.40	8.50	37.99	65.40	36.70	9.59	**42.72**	69.16	**40.13**	10.86
PointPillars [25]	*None*	30.85	52.45	27.31	5.59	34.09	59.88	32.80	8.52	38.24	64.02	35.76	8.01
	Fog [16]	30.39	52.13	26.79	5.71	35.38	60.81	35.15	9.60	37.74	64.56	34.48	7.30
	DROR [6]	29.32	**54.52**	21.88	4.82	30.99	57.17	28.43	6.95	34.72	60.59	30.34	6.72
	LISA [22]	28.70	49.78	24.98	5.63	33.87	60.93	31.38	8.70	37.92	63.98	34.61	7.94
	Ours-snow	**32.94**	54.21	**29.79**	**7.81**	**35.96**	**61.50**	**35.67**	**10.13**	**39.25**	64.37	**36.65**	**8.80**

The main experimental finding is that our simulation including consistently improves the performance on by a significant margin compared to both the baseline approach as well as all competing simulation [9], [21] and de-noising [3] methods, while not sacrificing but rather improving performance on clear weather as well.

Using simulation methods designed for different adverse conditions, such as the fog simulation in [24], does not transfer well to snowfall as the respective physical models differ; performance of [24] is slightly lower than the clear-weather baseline on both snowfall splits for most detection methods.

The application of DROR [3] as an enhancement step removing clutter points achieves among the lowest results, because it also removes several valid points, which do not belong to the snowfall clutter.

Qualitative results showing the proposed data augmentation scheme are presented in Figure 3.8. Here, PV-RCNN [17] is compared to the clear-weather baseline with no augmentation, DROR [3] and LISA [9]. In the first row, we see that the pedestrian inside the snowfall clutter can only be detected when our proposed data augmentation is applied during training. In the second row, additional false positives appear for all competing approaches. The bottom row shows a difficult highway scene with whirled-up snow dust. Our data augmentation approach generalizes well to this example, being the only method that detects the lead vehicle. Note also that in such a scenario with whirled-up snow dust, DROR [3] cannot remove the clutter completely.

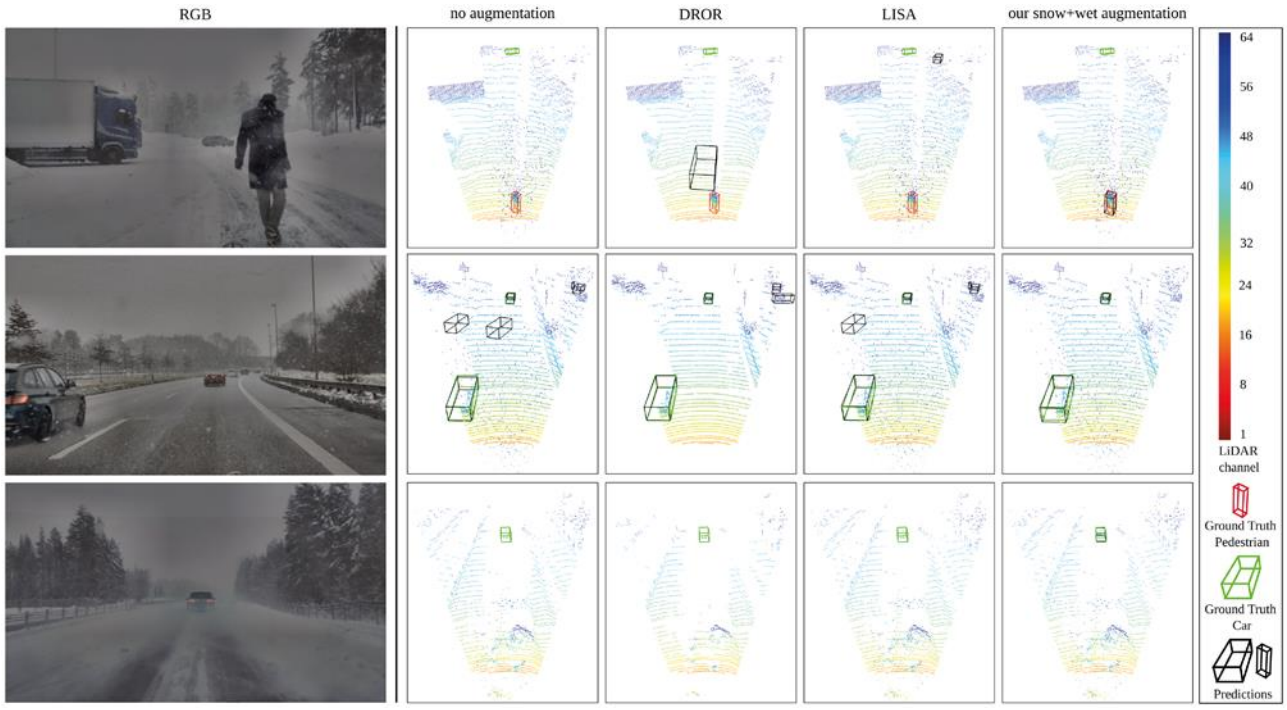

Figure 3.8: Qualitative comparison of PV-RCNN [20] on samples from STF [1] containing heavy snowfall. The leftmost column shows the corresponding RGB images. The rest of the columns show the LiDAR point clouds with ground-truth boxes and predictions using the clear-weather baseline (“no augmentation”), DROR [3], LISA [10], and our fully-fledged simulation (“our snow augmentation”).

3.4 Conclusion

In this Section, we have introduced a novel method for realistic synthesis of winter scenes from clear LiDAR captures modelling snowfall in a physically accurate way. Further, we have proven the effectiveness of the proposed algorithm, testing the augmentation with seven different 3D object detection methods and achieving consistent improvements of up to 2.1% in AP in heavy snowfall. As future work, we envision the exploration of temporal cues for robust LiDAR-based 3D object detection. For more information, you can have a look at our published result in [25].

4 Signal Enhancement

Converting RGB captures taken in adverse weather to clear daytime scenes is an effective way to broaden the scope of current perception algorithms, which are mainly trained using clear weather data. On the other hand, such a problem is intrinsically difficult due to its ill-posed nature and lack of data. In fact, adverse weather conditions follow a long-tail distribution where such environments are rarely encountered during day-to-day driving.

A straightforward solution could be to treat this task as a fully-supervised domain translation problem, where a deep neural network is trained to remove the adverse weather effects of the scene and output the dehazed capture. However, such approaches are limited by the lack of supervised data. In fact, given the dynamic nature of real-world automotive scenes, it is largely unfeasible to collect large-scale paired perturbed and clear data. It is possible to generate such paired data in a controlled environment like fog chambers, but this data would be limited in diversity and not representative of the real world, hindering the performances of any algorithm trained solely on this data.

Another method that could be employed in order to still use a fully-supervised approach is to create paired data by synthetically generate adverse weather conditions in clear weather scenes. However, also this approach is limited in performances as the descattering algorithm would mainly learn only the inverse of the (approximated) adverse-weather synthetization function used, rather than the different real-world effects of the adverse weather.

Hence, it is of great importance to use a (semi) unsupervised approach, capable of leveraging the unpaired clear and adverse weather data sources available. In this area, some of the best performing approaches are based on the CycleGAN intuition [26] of using multiple Generative Adversarial Networks to obtain a weakly-supervised adversarial training setting.

For example, Bijelic and colleagues recently proposed ZeroScatter [27], an algorithm capable of removing scattering media present in adverse weather conditions while only being trained in a weakly-supervised approach, with no direct supervision.

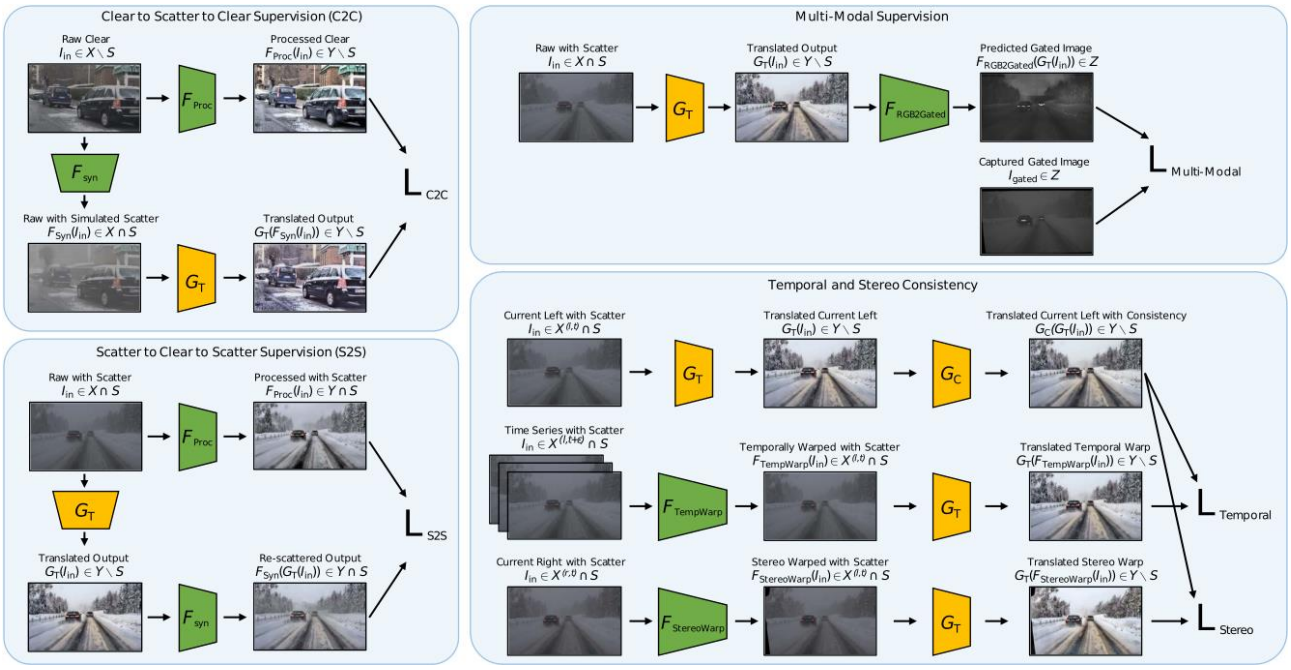

Figure 4.1: Overview of the ZeroScatter algorithm training process. On the left, the two cycle GANs training loop, while on the right the multi-modal, temporal and stereo consistency losses are illustrated.

In particular, ZeroScatter relies on different training cues:

1. Temporal: enforcing a consistency between adjacent frames of a video recording
2. Multi-view: exploiting the geometry transformation proprieties of stereo camera set-up
3. Multi modal: utilizing the additional supervision given by the output of a gated camera
4. Model based: by including the Koschmieder model for the synthetic fog generation.

Such novel combination of training cues promotes high-contrast, scatter-free, jitter-free results on unseen real-world scenes. A model-based supervision is employed using cycle training, which is facilitated by a robust adverse weather model, multi-modal supervision in the form of gated images for training on real heavy weather scenes, and consistency supervision in the form of temporal and stereo losses.

As shown in Figure 4.2, ZeroScatter significantly reduces the scattering present in the scene and reveals object in long distance, such as the house and trees in the top two examples below. Compared to EPDN and PFF-Net, ZeroScatter is able to produce images with better contrast and less noise.

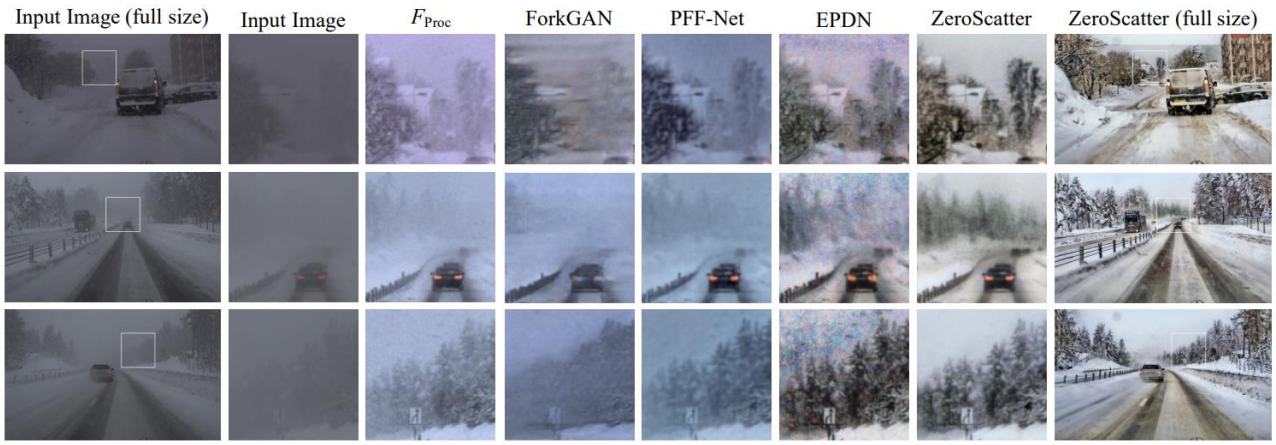

Figure 4.2: Real-world data qualitative comparisons against state of the art approaches for image descattering.

It is important to note that the original version of the Koschmieder model assumes the air light to be constant across the whole image. While this simplification could be acceptable in broad daylight conditions, it does not hold true anymore when there are visible active light sources, especially at dusk or night. To improve such model, in ZeroScatter the different light sources are first detected and then their spreading into the scattering media is modelled.

In particular, the light sources are estimated by applying a Top-hat filter on the radiance channel and then all objects above the air light target value (estimated using a reparametrized dark channel prior) are filtered. The light scattering is then modelled by applying a Gaussian blur.

While such modelling is an efficient and straightforward way to represent such effect, it is just a simple approximation which is not physically accurate and hence might notably differ from the real-world phenomenon.

5 Sensor Enrichment

5.1 Depth Prediction

Accurate three-dimensional environment perception constitutes a cornerstone for driver assistance systems to ensure safety and comfort for both drivers and passengers. To achieve accurate perception, several sensors are required, which supply the vehicle with measurements of the surroundings and enable the construction of an environment model. These vehicle perception sensors can be divided into active and passive types. Most cameras are referred to as passive sensors since image information is captured passively, using only light emitted by the sun and external light sources such as driving lights. In contrast, active sensors like light detection and ranging (LiDAR) and radio detection and ranging (RaDAR) illuminate the scene with specific radiation and capture the reflections of the detected objects. The benefit of active sensors is that the quality of 3D sensing is independent of external light sources, resulting in accurate sensor measurements even at night or in low lighting conditions.

5.1.1 Depth from Wide Baseline Stereo using RGB images

In stereo-based depth estimation, a pair of images taken from two different viewpoints are used for 3D reconstruction. The two images are typically captured with a pair of synchronized cameras with a horizontal displacement between their center of projection. In a wide baseline setup, the horizontal displacement is typically large enough to create very different views of the scene. The key problem in stereo is finding dense pixel correspondences between the two images. One of the challenges in wide baseline stereo for automotive use cases is that large part of the scene can be monocular and exact correspondence is not possible. In traditional small baseline stereo vision, the pair of images are rectified to constraint the correspondence search along a single dimension. The image rectification process warps both images into a common image plane such that they only differ by a horizontal displacement $d(u, v)$, for each image coordinate (u, v) in the *reference* image. The horizontal displacement d is also known as disparity and using the known focal length of the camera f and the baseline b, the corresponding depth z is obtained by $z = \frac{fb}{d}$. The main goal of binocular stereo is then to find the disparity function $d(u, v)$ that matches the *left-right* image coordinates (pixels). The typical stereo pipeline starts with finding corresponding points in both images and then a matching cost is formed. In general, the maximum distance of binocular stereo is fundamentally limited by the baseline b, focal length f, and the observation noise. However, in wide baseline stereo with long focal length small calibration errors can result in inaccurate rectification and therefore requires extracting robust rotation invariant features.

To enable perception at far distances, the AI-SEE sensor setup includes a wide baseline stereo system with high resolution images. Figure 5.1 shows the wide baseline stereo setup and Figure 5.2 shows long range depth estimation results, using stereo images captured with this setup as input, at day and night scenarios.

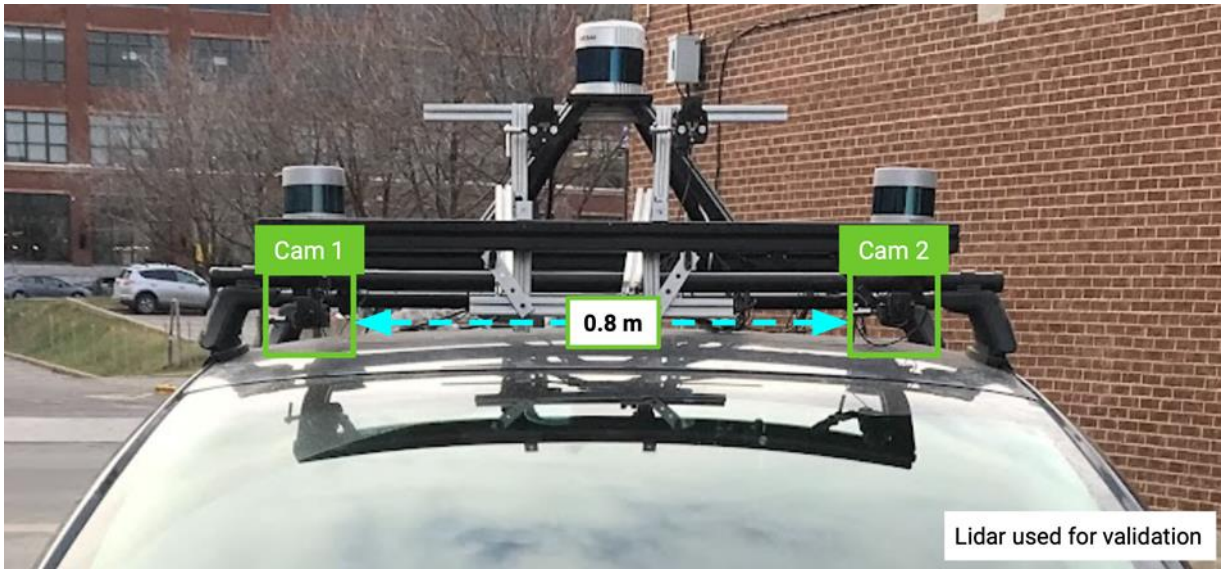

Figure 5.1: Wide baseline stereo cameras setup. Left and right cameras are positioned at a 0.8m distance.

Figure 5.2: Wide baseline stereo depth estimation results in day and night lighting conditions.

The proposed depth estimation neural network architecture (see Figure 5.3) consists of differentiable image processing layers operating on the pair of HDR raw images and jointly optimized with the disparity estimation network. The differentiable raw processing pipeline includes linearization, white balance, color correction, demosaicing, HDR tone mapping, de-noising, sharpening, dark channel prior based dehazing and diffuse-specular separation network. This allows the output of the trainable image-processing module to produce high contrast features that are robust to illumination variation, fog, shadow, and ground reflection. The output of the differentiable image processor is then fed to a pose refinement network, which accounts for the calibration noise as well as frame-to-frame misalignment due to vibration and produces rectified feature maps. A multiresolution feature encoder then extracts features robust to rotational variations. The encoder output from the two images are then correlated to build a 4D cost volume with disparity as the additional dimension. The cost volume is then processed by the decoder network and the final disparity is estimated per-pixel.

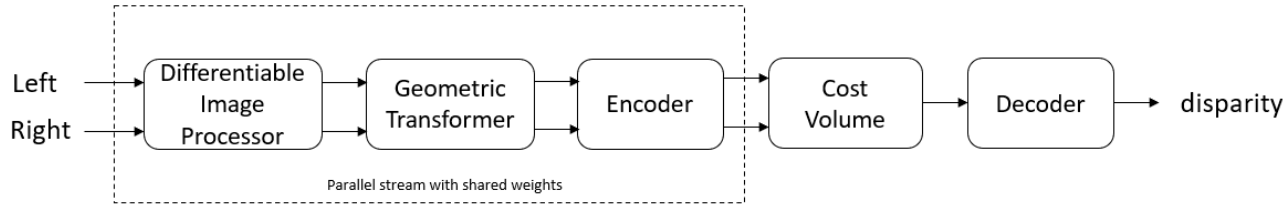

Figure 5.3: Wide baseline stereo Network scheme. Instead of conventional ISPs, our model includes differentiable image processing layers that are optimized end-to-end along with the disparity estimation network.

The model is trained using a combination of supervised and semi-supervised approach. For supervised training, we use synchronized LiDAR captures as well as spatio-temporally aggregated LiDAR data. However, LiDAR data is sparse and limited in depth. To account for that we generate high-resolution dense pseudo disparity maps using an ensemble of stereo models. Furthermore, we introduce spatio-temporal consistency training losses and domain transfer from synthetic to real depth prediction. Specifically synthetic data is used to improve robustness to reflective regions (e.g. wet regions), which is one of the most challenging cases to handle in this depth prediction task.

5.1.2 Gated2Gated: Self-Supervised Gated Depth Estimation

Previous neural network based gated depth estimation methods require LiDAR supervision for predicting depth maps from multiple gated images. However, LiDAR ground truth offers only sparse spatial resolution and suffers from backscatter in adverse weather. Furthermore, current LiDAR systems only provide reliable measurements for ranges of up to 100m. This limits the gated depth estimation algorithms to the restrictions of the LiDAR system.

For this reason, we have developed a self-supervised gated depth estimation method within the AI-SEE project, which does not require any supervision for training the neural network. The self-supervision allows to overcome the limited depth range of methods trained on LiDAR ground-truth and removes complex synchronization processes between LiDAR and cameras. Furthermore, we can train our model on harsh weather conditions, e.g., fog, rain, or snow, where LiDAR-based ground-truth is not available.

The proposed method takes three gated images as input and predicts the scene albedo, depth, and ambient illumination. By reconstructing the input gated images with the predictions and the pre-defined range-intensity-profiles (RIPs), and by using temporal depth information, we train the neural network in a self-supervised training scheme.

5.1.2.1 Method

The proposed network architecture, called Gated2Gated, is illustrated in Figure 5.4. The network takes three gated images Z_t^i, for $i = \{0,1,2\}$ as input, at each time t. The gated measurements Z_t^i are concatenated in a tensor Z_t that is fed to three different convolutional neural networks that disentangle the input into albedo, ambient light, and depth. These predictions are then used to reconstruct the input images using a cyclic loss. In addition to this novel gated imaging-based training signal, we exploit temporal consistency between temporally adjacent gated frames to handle regions with shadows and multi-path reflections.

Specifically, the proposed architecture is composed of three networks. The first network predicts a dense depth map per gated tensor Z_t, denoted as $f_r: Z_t \rightarrow \hat{r}_t$. The second network also takes Z_t as input, and predicts ambient and albedo, denoted as $f_{\Lambda\alpha}: Z_t \rightarrow (\widehat{\Lambda_t}, \widehat{\alpha_t})$. The third network takes two temporally adjacent gated tensors as input (Z_t, Z_n), and predicts a rigid 6 DoF pose transformation $\hat{X}$ from Z_n to Z_t, denoted as $\widehat{X_n} = \begin{pmatrix} R_{t\rightarrow n} & t_{t\rightarrow n} \\ 0 & 1 \end{pmatrix}$, with $R_{t\rightarrow n} \in SO(3)$ and $t_{t\rightarrow n} \in \mathbb{R}^{3x1}$ generated by $f_{t\rightarrow n}: (Z_t, Z_n) \rightarrow \hat{X}_{t\rightarrow n}$.

The learned function f_r is optimized to predict the absolute depth value and is supervised using the other two auxiliary functions, $f_{\Lambda\alpha}$ and $f_{t\rightarrow n}$. The first auxiliary function is used to exploit cyclic measurement consistency with the measured gated slice, i.e., enforce that the predicted depth is consistent with a gated measurement. The second auxiliary function allows us to exploit temporal consistency between nearby gated frames. Using these cues, the proposed method resolves scale ambiguity inherent in monocular depth estimation. The two consistency components are discussed in the following sections.

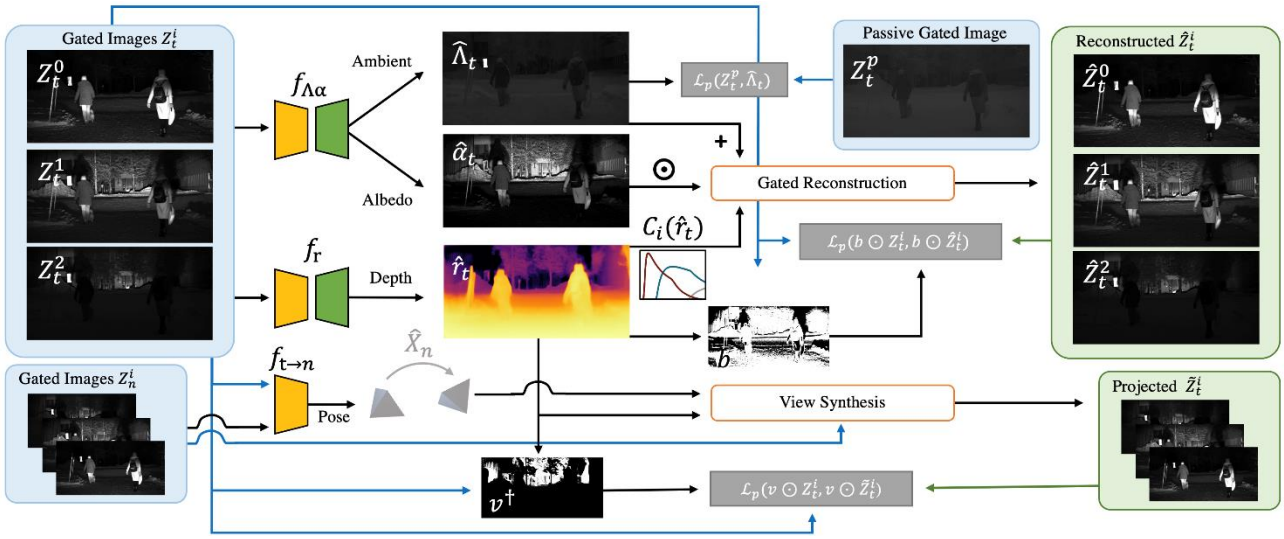

Figure 5.4: Network architecture of the proposed self-supervised gated depth estimation approach. The network estimates dense depth from a set of three gated images by learning from cycle gated and temporal consistency.

5.1.2.1.1 Cycle Gated Consistency

The cyclic gated consistency loss supervises the learning process of the predicted depth $\hat{r}$, the ambient illumination $\widehat{\Lambda}$ and the albedo $\hat{\alpha}$ by reconstructing the gated slices and comparing them to the input gated images. To reconstruct the gated images we use

$$\widehat{Z_t^i} = \hat{\alpha}_t C_i(\hat{r}_t) + \widehat{\Lambda}_t$$

With C_i being the range-intensity profiles. The C_i profiles 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 $N = 6$. The ambient prediction $\widehat{\Lambda}_t$ can be directly supervised using the ground truth captured passive images Z_t^p and the photometric loss $\mathcal{L}_p$. As loss function $\mathcal{L}_p$, we use Structural Similarity (SSIM) [28] and $\mathcal{L}_1$ norm, that is

$$\mathcal{L}_p\left(Z_t^i, \hat{Z}_t{}^i\right) = 0.85\frac{1-SSIM(z_t^i, \hat{z}_t{}^i)}{2} + ||z_t^i - \hat{z}_t{}^i||_1.$$

Using the previous equations allows to reconstruct the input gated images and train the network self-supervised. However, adopting a photometric loss between the measurement and the prediction fails in practice as severe multipath effects, missing illumination due to occlusion, and saturation due to retro-reflective signs can break the model assumptions. Therefore, we introduce a pixel-based mask b that remove these inconsistent areas during the training process.

The final Gated2Gated cyclic loss function is then defined as

$$\mathcal{L}_{cyc} = \sum_{i=0}^{2} \mathcal{L}_p\left(b \odot Z_t^i, b \odot \hat{Z}_t{}^i\right) + \mathcal{L}_p(\widehat{\Lambda}_t, Z_t^p)$$

5.1.2.1.2 Temporal Depth Consistency

As illustrated in Figure 5.4, we use view synthesis to introduce temporal consistency between adjacent gated images during training. Specifically, we reconstruct the view of central gated image Z_t from temporal neighbors Z_n using the camera matrix K, the predicted depth $\hat{r}_t$ and camera pose transformation $\hat{X}_{t\to n}$. Considering x_t and x_n homogeneous (pixel) coordinates from Z_t^i and Z_n^i, the mapping from source pixels x_t to target pixels x_n is defined as follows

$$x_n \sim K\,\hat{X}_{t\to n}\,\hat{r}_t K^{-1} x_t$$

We compare the reconstructed view $\hat{Z}_t$ with Z_t using photometric loss and use it to train the depth prediction network f_r. Unfortunately, this naive approach fails in the presence of moving occlusions due to ego-motion and the movement of non-stationary objects, which violate the rigid pose transformation. As earlier for the cyclic loss, we introduce a validity mask v. Additionally, scene points that are only visible either in the source, or the target image break our image formation model. Such scenarios can be caused by occlusions in the foreground, obstructing the view of the background. In dynamic scenes, this occlusion changes in each time step. For a triplet of adjacent frames $(t-1, t, t+1)$, we can define the minimum pixel error out of those pairwise differences, as occlusions cause a higher re-projection error. This can be explained by the fact, that the obscuring object in the foreground and the background have greater differences in texture than neighbouring pixels in the background. Therefore, we calculate minimum of per-pixel loss between the re-projection from two temporal adjacent pairs as

$$\mathcal{L}_{temp} = \min_{n=\{t-1,t+1\}} \sum_{i=0}^{2} \mathcal{L}_p\left(v \odot Z_t^i, v \odot \hat{Z}_t{}^i(n)\right)$$

The complete loss function is defined by

$$\mathcal{L} = \mathcal{L}_{temp} + \lambda_{cyc}\mathcal{L}_{cyc},$$

where $\lambda_{cyc} = 0.01$ is determined empirically.

5.1.2.2 Dataset

In order to train our proposed models, we collected 1835 video sequences, which comprised about 130,000 frames. Each time history is centered around one of the 13000 middle frames and provides a temporal history of 1s at a sampling rate of 10Hz. The center frames are preselected by human annotators depending on the scene of interest from an underlying data distribution covering diverse winter road scenes collected in Northern Europe.

We evaluate our proposed method on the open-source Gated2Depth [29] and SeeingThroughFog [1] datasets. While the first dataset contains a large variety of day and night captured images, the second contains diverse cluttered recordings in light fog, dense fog and snowfall conditions, which allow us to evaluate the performance of our model in harsh weather conditions and scenarios where obtaining ground truth data is difficult. In the last case, to filter out clutter from LiDAR ground truth, we use the DROR filtering algorithm [3].

5.1.2.3 Results

We compare the performance of the proposed Gated2Gated method against state-of-the-art supervised and self-supervised depth estimation methods. As supervised approaches, we compare against gated depth estimation [29] [30] LiDAR depth completion [31] and stereo vision [32] [33] . For comparison with unsupervised methods, we consider stereo [34] and temporal based self-learning approaches [35] [36]. Since the self-supervised baseline methods do not provide absolute depth predictions, we scale the estimated depth maps with median ground-truth LiDAR information. We evaluate using the metrics RMSE, MAE, ARD and $\delta_i < 1.25i$ for $i \in \{1,2,3\}$. These metrics are computed for distances between 3m and 80m, limited by the maximum LiDAR distance. To evaluate the long range influence in adverse weather we rely on 7m bins.

5.1.2.3.1 Evaluation on Clear Data – Gated2Depth Dataset

Table 5.1 reports a quantitative comparison of our proposed Gated2Gated method and other state-of-the-art methods on the test set of the Gated2Depth dataset [29]. Our model outperforms all other self-supervised methods [34] [35] [36], and even temporal approaches [35] [36] that use LiDAR ground truth for depth scaling. The proposed method also outperforms stereo [32] [33] and Bayesian-based gated depth estimation methods [30]. Among the supervised methods, only Sparse-to-Dense [31] and Gated2Depth [29] obtain better results. Figure 5.5 qualitatively compares our method to baseline methods for depth estimation. While the performance metrics for Gated2Depth are on par with the proposed method, the qualitative comparison shows that Gated2Gated predicts much finer grain details and sharper object contours in depth maps. Furthermore, the proposed method generalizes better to far distances than the other methods.

Table 5.1: Comparison of the proposed framework and state-of-the-art methods on the Gated2Depth test dataset. We compare our model to supervised and unsupervised approaches. M refers to methods that use temporal data for training, S for stereo supervision, G for gated consistency and D for depth supervision.*marked methods are scaled with LiDAR ground truth.

	Method	Modality	Train	RMSE [m]	ARD	MAE [m]	δ_1 [%]	δ_2 [%]	δ_3 [%]	Compl. [%]
	Real Data – Night (Evaluated on Lidar Ground Truth Points)									
Supervised	PSMNet [10]	Stereo-RGB	D	14.58	0.21	8.34	68.75	82.63	89.36	100
Supervised	SGM [29]	Stereo-RGB	-	15.51	0.36	8.75	63.94	76.19	82.31	63
Supervised	Sparse-to-Dense [43]	Lidar(GT)+RGB	D	8.79	0.21	4.38	**87.64**	**93.74**	**95.88**	100
Supervised	Regression Tree [1]	Gated	D	10.54	0.24	6.01	76.73	89.74	93.45	40
Supervised	Least Squares	Gated	-	13.13	0.42	8.88	43.60	55.80	63.54	31
Supervised	Gated2Depth [22]	Full-Gated	D	14.86	0.29	8.84	58.79	58.79	79.84	100
Supervised	Gated2Depth [22]	Gated	D	**8.39**	**0.15**	**3.79**	87.52	93.00	95.21	100
Unsupervised	Monodepth [17]	RGB	S	11.41	0.23	6.18	76.64	89.53	94.19	100
Unsupervised	Monodepth [17]	Full-Gated	S	15.41	0.52	11.33	31.72	71.23	88.74	100
Unsupervised	*Packnet [24]	RGB	M	12.15	0.27	6.87	69.14	86.93	92.57	100
Unsupervised	*Packnet-Slim [24]	Gated	M	10.78	0.22	6.02	74.37	89.44	94.34	100
Unsupervised	*Monodepth2 [19]	RGB	M	14.92	0.38	9.98	39.85	68.57	83.99	100
Unsupervised	*Monodepth2 [19]	Gated	M	11.18	0.25	5.99	76.79	87.04	91.58	100
Unsupervised	**Gated2Gated**	Gated	MG	**9.43**	**0.21**	**4.86**	**82.17**	**91.54**	**94.48**	100

	Method	Modality	Train	RMSE [m]	ARD	MAE [m]	δ_1 [%]	δ_2 [%]	δ_3 [%]	Compl. [%]
	Real Data – Day (Evaluated on Lidar Ground Truth Points)									
Supervised	PSMNet [10]	Stereo-RGB	D	13.94	0.19	7.78	71.32	84.67	91.38	100
Supervised	SGM [29]	Stereo-RGB	-	9.63	0.17	4.59	85.80	92.72	95.20	86
Supervised	Sparse-to-Dense [43]	Lidar(GT)+RGB	D	8.21	0.16	4.05	**88.52**	**94.71**	**96.87**	100
Supervised	Regression Tree [1]	Gated	D	15.83	0.49	11.40	56.30	75.54	82.45	23
Supervised	Least Squares	Gated	-	19.52	0.75	14.05	43.42	54.63	63.76	16
Supervised	Gated2Depth [22]	Full-Gated	D	13.75	0.26	8.16	62.48	62.48	82.93	100
Supervised	Gated2Depth [22]	Gated	D	**7.61**	**0.12**	**3.53**	88.07	94.32	96.60	100
Unsupervised	Monodepth [17]	RGB	S	10.24	0.18	5.47	80.49	91.78	95.61	100
Unsupervised	Monodepth [17]	Full Gated	S	13.33	0.40	9.51	36.64	81.63	92.86	100
Unsupervised	*Packnet [24]	RGB	M	12.44	0.27	7.23	66.32	85.85	92.40	100
Unsupervised	*Packnet-Slim [24]	Gated	M	9.93	0.18	5.34	78.98	91.83	96.06	100
Unsupervised	*Monodepth2 [19]	RGB	M	13.18	0.33	8.79	44.99	75.87	90.65	100
Unsupervised	*Monodepth2 [19]	Gated	M	9.57	0.18	4.76	83.20	91.75	94.94	100
Unsupervised	**Gated2Gated**	Gated	MG	**8.46**	**0.17**	**4.37**	**83.56**	**93.12**	**96.09**	100

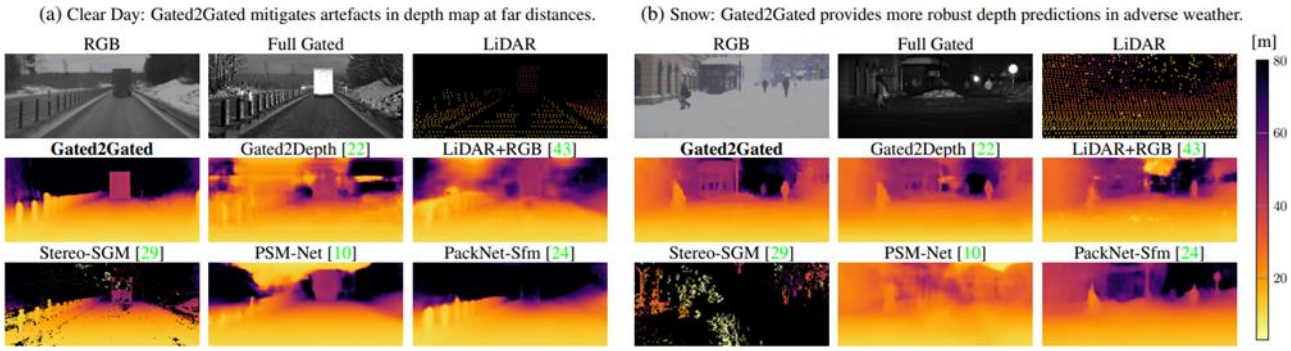

Figure 5.5: Qualitative comparison of the proposed Gated2Gated and existing methods.

5.1.2.3.2 Evaluation on Adverse Weather Scenes – Seeing Through Fog Dataset

We also evaluate the proposed method in adverse weather, adopting the test splits provided in [1]. The performance is measured in binned metrics to weight all distances equally. Table 5.2 shows the quantitative results of the Gated2Gated method and state-of-the-art methods. We note that absolute metrics may improve in adverse weather conditions, as the number and range of ground-truth LiDAR points decrease with worse weather conditions. We validate that Gated2Gated achieves robust performance overall weather conditions. In contrast, Monodepth2 and Sparse-to-Dense struggle to maintain performance in adverse weather. Since Sparse-to-Dense uses LiDAR points as additional inputs, wrong depth measurements from backscatter negatively impact the predicted depth maps. Furthermore, Table 5.2 validates that the proposed approach performs on par with Gated2Depth, and for daytime as well as in harsh weather scenarios, Gated2Gated even outperforms Gated2Depth.These results highlight the generalization capabilities of the proposed method over a wide range of distances and weather conditions.

Table 5.2: Evaluation of the proposed Gated2Gated framework and state-of-the-art-methods on adverse weather scenes. All metrics are evaluated in bins of approximately 7m to weight all distances equally. G indicates training and evaluation on gated images.

	Method	clear					light fog					dense fog					snow				
		RMSE	MAE	δ_1	δ_2	δ_3	RMSE	MAE	δ_1	δ_2	δ_3	RMSE	MAE	δ_1	δ_2	δ_3	RMSE	MAE	δ_1	δ_2	δ_3
Day	Monodepth RGB [17]	12.74	8.43	75.11	90.18	**94.81**	14.04	9.10	72.70	88.43	94.32	14.67	10.64	63.49	82.90	91.89	13.17	8.73	71.56	89.06	**94.81**
	Sparse-to-Dense [43]	13.66	9.85	54.20	82.42	91.47	14.23	10.66	49.75	79.62	90.10	18.50	15.35	37.04	64.67	78.25	13.42	9.81	53.12	82.29	92.04
	Packnet-Slim G [24]	16.46	11.62	56.91	78.43	88.48	16.95	11.80	59.09	78.80	88.81	17.01	12.09	54.93	76.37	88.89	15.30	10.33	62.22	82.29	90.65
	Monodepth2 G [19]	13.26	7.40	78.59	88.95	92.77	18.17	10.43	72.91	83.20	89.27	15.56	8.72	76.79	85.38	90.68	12.84	7.12	**80.04**	89.34	93.13
	Gated2Depth [22]	11.48	6.60	79.17	87.38	91.58	11.28	6.63	81.20	88.66	92.56	11.86	7.85	71.72	87.10	91.70	11.28	6.61	78.87	87.93	92.50
	Gated2Gated	**11.15**	**6.31**	**80.82**	**90.48**	93.97	**10.70**	**6.01**	**84.71**	**91.52**	**94.65**	**11.09**	**6.86**	**81.09**	**91.43**	**94.47**	**10.97**	**6.28**	80.01	**91.12**	94.63
Night	Monodepth RGB [17]	13.78	8.92	72.63	88.48	93.37	13.30	8.75	72.52	88.88	**94.45**	16.31	10.99	69.15	85.92	**91.33**	15.28	9.89	68.88	86.86	**93.44**
	Sparse-to-Dense [43]	14.43	10.40	50.32	78.66	89.58	13.92	10.01	51.88	80.41	90.70	16.54	12.07	47.15	73.45	84.36	14.08	10.05	52.91	81.03	90.85
	Packnet-Slim G [24]	15.81	11.11	59.80	79.52	88.65	16.01	11.23	58.44	80.60	89.57	17.49	12.60	57.72	77.77	87.26	16.47	11.40	60.17	79.55	88.62
	Monodepth2 G [19]	14.52	8.30	74.45	85.41	89.73	14.21	8.29	74.56	85.06	91.01	18.33	11.88	66.61	79.25	84.48	15.11	8.46	76.15	86.38	90.52
	Gated2Depth [22]	**10.06**	**5.17**	**84.81**	**90.59**	**93.39**	**9.94**	**5.37**	**81.95**	**89.80**	93.63	**12.51**	**7.72**	**76.90**	86.59	90.81	**10.70**	**5.81**	**81.81**	89.45	93.02
	Gated2Gated	11.69	6.74	80.25	89.58	92.83	11.29	6.46	79.39	89.31	93.17	13.52	8.69	76.43	**86.70**	90.61	11.91	6.80	80.76	**90.09**	93.31

5.2 Parameter Optimization

5.2.1 Optimization of the Gating Parameters

Currently, the gated camera is operated with handcrafted gating parameters that determine the camera output. To exploit the full capabilities of the gated camera, these parameters need to be optimized for specific tasks (see Deliverable 4.1 “Optimization of the Gating Parameters”).

The behaviour of the gated camera is configurable according to a set of user-adjustable hyper-parameters, however the details of its inner-workings are not revealed to the user [37]. Therefore, the gated imaging system can be considered as a black-box unit.

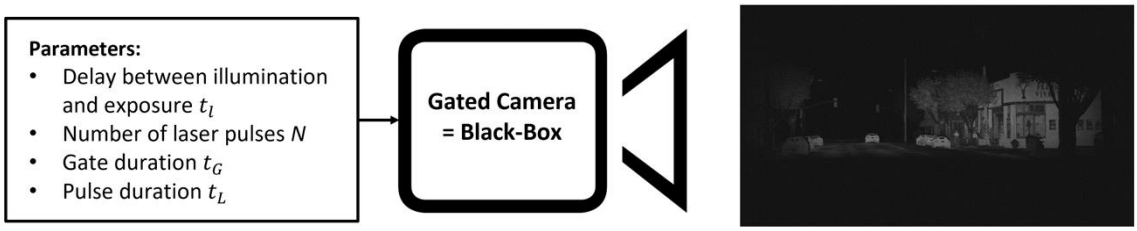

Figure 5.6: Configurable parameters of a black-box Gated camera.

As shown in Figure 5.6, the output image of the gated camera can be influenced by adjusting the delay between illumination and exposure t_l, the number of laser pulses N, the gate duration t_G and the pulse duration t_L. These parameters determine the range-intensity-profiles which represent the distance-dependent intensity values of the output image and therefore which distance of the scene is illuminated.

The reconstruction of the depth is generally done with a neural-network-based approach that uses multiple gated images of the same scene with different range-intensity-profiles. The quality of those depth-maps is heavily influenced by what range-intensity-profiles are used. So far, the profiles are selected manually by considering the prior knowledge that the depth can be estimated reliably by using overlapping profiles that cover different ranges. However, the range-intensity-profiles were not refined or optimized in any way

since that would have involved a manual exhaustive search with adjusting the gating parameters and collecting a high amount of data.

In a first step, we intent to develop an optimization algorithm that automatically adjusts the gated parameters for the specific task of learning the depth with the environmental conditions night and clear weather. This algorithm should later be used to optimize the parameters for other tasks and environmental conditions as well.

The algorithm follows the approach of [37] that uses a neural network as a differentiable proxy function to perform the optimization in two stages.

5.2.2 Method

In the first stage, the proxy function is trained to learn the mapping between gating parameters and camera output. In other words, the black box that represents the gated imaging system is represented by a differentiable neural network to allow the recreation of gated images with specific gating parameters. In the second stage, the proxy function is fixed and used to generate a high number of gated images with varying parameters to find the optimal settings for the depth estimation.

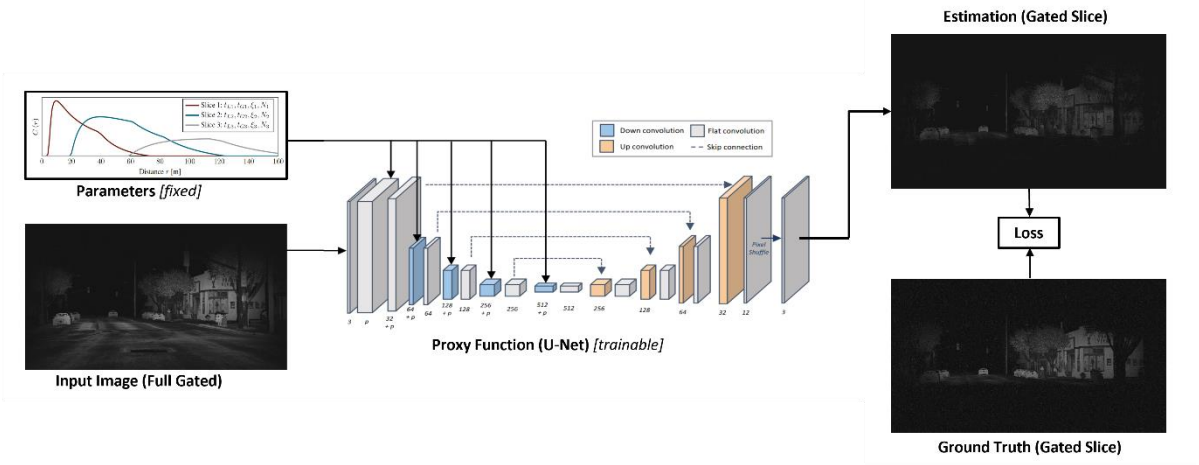

Figure 5.7: Architecture of stage one - Training of the differentiable proxy function.

The architecture of the first stage (Figure 5.7) shows that a convolutional neural network is used to approximate the black box. A loss that is determined by comparing the estimated gated image with the ground truth gated image is minimized by an optimizer. The differentiable proxy function is trained to reproduce the entire black box process as a function of its input configuration parameters.

In practice, we use a variant of the UNet CNN architecture with an encoder-decoder structure. In addition to the input image channel, we concatenate as many channels as there are gated parameters. Each channel holds the value of the hyperparameter replicated over the spatial dimension. To encourage the model to learn the effects of the parameters on the output image, we append the input hyperparameter channels to each downsample-layer (shown as blue layers in Figure 5.7). We train the model using a limited number of handcrafted parameters.

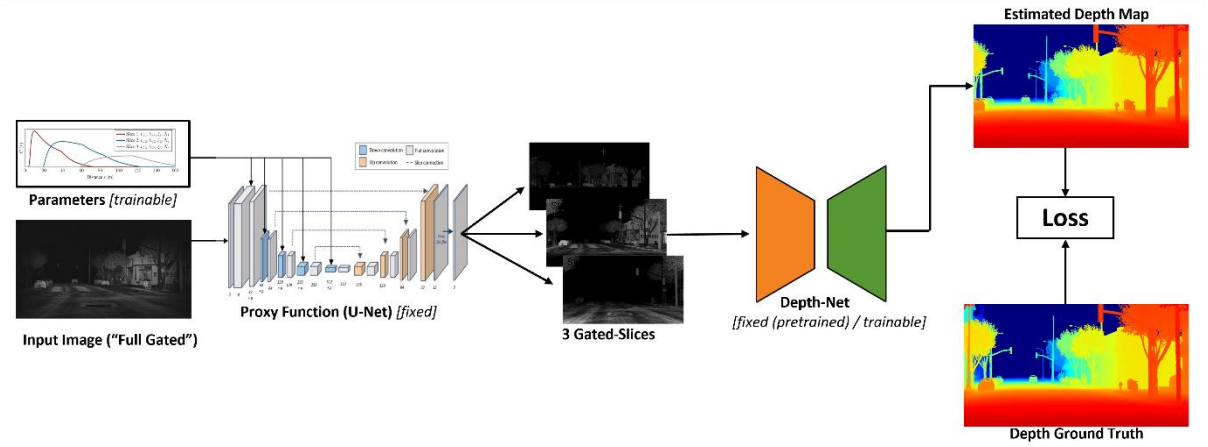

Figure 5.8: Architecture of Stage 2 - Fixed Proxy Function is used to generate gated images for training a neural-network-based depth estimation approach.

In stage 2, the trained differentiable proxy function is fixed and used to optimize the gated parameters as shown in Figure 5.8. By using the fixed proxy function, an arbitrary number of gated images with various gating parameters can be created. These gated images can be used to train a second neural network, which is trained to fulfill a specific task. In the example of Figure 5.8, it is used to estimate the depth map, but a variety of other tasks like object detection or scene recognition would be conceivable. The neural network is trained to minimize a loss that can be derived by comparing the estimated output with the task-specific ground-truth.

Since both the proxy function and the neural network used for depth estimation are differentiable, an optimizer algorithm can perform backpropagation through all stages, which allows an end-to-end optimization of the gating parameters while training the neural-network-based depth estimation approach.

6 Conclusion

The second deliverable of work package *WP4 Sensor Fusion and AI* was dedicated to the enhancement and enrichment of raw sensor data streams affected by adverse weather conditions using novel generative neural network (NN) models.

In this deliverable, we first described a signal enhancement method that we used to improve the detection results of LiDAR systems by up to 17% in the "snow" adverse weather case. The basis for this improvement was a sufficiently large sample of learning data, which we were able to provide in this case via a simulation method we developed in the project, which we have described in detail in this paper.

We then presented another example of a signal enhancement method with the "ZeroScatter" approach, a defogger for image data. This “ZeroScatter” network was trained predominantly on simulated fog images and was shown to perform highly when applied to real fog images. The “ZeroScatter” approach was about seeing the effect of a defogger and optimizing its performance. This approach will later serve as a precursor to a detector that also works well in fog. The detection performance of a standard detector has increased by up to 20% over standard detectors in initial measurements after applying this preliminary stage in fog.

In the final chapter of this deliverable, we report on the progress in the development of two methods for generating a depth map: Depth estimation using a newly developed deep neural network for a wide-base stereo system, and depth estimation using depth slices from a monocular gated camera.

The deep neural network for generating a depth map from wide-base stereo camera images realizes optimization of all components throughout the processing chain, i.e., from the lens with its fixed parameters to the detection task. This optimization causes the maximum performance to be achieved over the entire processing chain. This work is a first step towards solving the technically very challenging and currently still unsolved lost-cargo problem, i.e. the detection of small objects at large distances (up to 150m) even under poor weather and visibility conditions.

In further developing depth estimation using a monocular gated camera, we addressed self-supervised depth calibration. This self-supervised depth calibration allows to make depth estimation much finer and also to extend the coverage of the depth map significantly. It represents another step towards solving the lost cargo problem mentioned above. The next goal in the further development of the gated camera is to develop a method that realizes an automatic adaptation of it to the prevailing weather and visibility conditions in order to guarantee maximum performance in any weather and visibility condition. Furthermore, it is planned to extend the gated camera system as a gate camera stereo system in order to combine the optimization of the gated camera system with the results from the deep neural network work for the stereo camera approach.

List of abbreviations

ABBREVIATION	MEANING
NN	Neural Network
ADS	Advanced Driving Systems
STF	Seeing Through Fog
LiDAR	Light Detection and Ranging
DROR	Dynamic Radius Outlier Removal
RaDAR	Radio Detection and Ranging
HDR	High Dynamic Range
AP	Average Precision
LISA	Lidar Light Scattering Augmentation
RGB	Red Green Blue
RIPs	Range-Intensity-Profiles
SSIM	Structural Similarity

References

[1] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer and F. Heide, "Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather," IEEE, 2020.

[2] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czamecki and S. Waslander, "Canadian adverse driving conditions dataset.," in *The International Journal of Robotics Research*, 2021.

[3] N. Charron, S. Phillips and S. L. Waslander, "De-noising of Lidar Point Clouds Corrupted by Snowfall," IEEE, 2018.

[4] G. Williams, "Optimization of eyesafe avalanche photo-diode LiDAR for automobile safety and autonomous navigation systems.," in *Optical Engineering*, 2017.

[5] Q. Ding, W. Chen, B. King, Y. Liu and G. Liu, "Combination of overlap-driven adjustment and phong model for LiDAR intensity correction.," in *Photogrammetry and Remote Sensing*, 2013.

[6] A. Kashani, M. Olsen, C. Parrish and N. Wilson, "A review of LiDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration. Sensors, 15(11)," in *Sensors*, 2015.

[7] Velodyne, *Hdl-64e user's manual,* 2007.

[8] R. Rasshofer, M. Spies and H. Spies, *Influences of weather phenomena on automotive laser radar systems.,* vol. 9, 2011.

[9] V. Kilic, D. Hegde, V. Sindagi, A. B. Cooper, M. Foster and V. Patel, "LiDAR light scattering augmentation (LISA): physics-based simulation of adverse weather conditions for 3D object detection," in *https://arxiv.org/abs/2107.07004*, 2021.

[10] 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 Recognition*, 2012.

[11] B. Yang, W. Luo and R. Urtasun, "Pixor: Realtime 3D object detection from point clouds.," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] F. Julca-Aguilar, J. Taylor, M. Bijelic, F. Mannan, E. Tseng and F. Heide, "Gated3D: Monocular 3D object detection from temporal illumination cues.," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[13] A. Simonelli, S. Rota Bulo, L. Porzi, M. Lopez-Antequera and P. Kontschieder, "Disentangling monocular 3D object detection.," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[14] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3D object detection.," in *AAAI Conference on Artificial Intelligence*, 2021.

[15] A. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds.," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang and H. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[17] S. Shi, X. Wang and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] S. Shi, Z. Wang, J. Shi, X. Wang and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[19] Y. Yan, Y. Mao and B. Li, "SECOND: Sparsely embedded convolutional detection," in *Sensors*, 2018.

[20] T. Yin, X. Zhou and P. Krahenbuhl, "Center-based 3D object detection and tracking.," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] 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.

[22] J. Casselgren, M. Shödal and J. LeBlanc, "Angular spectral response from covered asphalt," in *Applied Optics*, 2007.

[23] OpenPCDet Development Team, "OpenPCDet: An open-source toolbox for 3D object detection from point clouds," 2020. [Online]. Available: https://github.com/open-mmlab/OpenPCDet. [Accessed 15 11 2021].

[24] M. Hahner, C. Sakaridis, D. Dengxin and L. Van Gool, "Fog simulation on real LiDAR point clouds for 3D," in *Computer Vision and Pattern Recognition*, 2021.

[25] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai and L. van Gool, "LiDAR Snowfall Simulation for Robust 3D Object Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] Z. Shi, E. Tseng, M. Bijelic, W. Ritter and F. Heide, "Domain Transfer for Long Distance Imaging and Vision through Scattering Media," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, no. 4, p. 600–612, 2004.

[29] T. Gruber, F. Julca-Aguilar, M. Bijelic and F. Heide, "Gated2Depth: Real-Time Dense Lidar From Gated Images," IEEE, 2019.

[30] A. Adam, C. Dann, O. Yair, S. Mazor and S. Nowozin, "Bayesian Time-of-Flight for Realtime Shape, Illumination and Albedo," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 5, p. 851–864, 2017.

[31] F. Ma and S. Karaman, "Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image," IEEE, 2018.

[32] J.-R. Chang and Y.-S. Chen, "Pyramid Stereo Matching Network," IEEE, 2018.

[33] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, p. 328–341, 2008.

[34] C. Godard, O. M. Aodha and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," IEEE, 2017.

[35] C. Godard, O. M. Aodha, M. Firman and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," IEEE, 2019.

[36] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos and A. Gaidon, "3D Packing for Self-Supervised Monocular Depth Estimation," IEEE, 2020.

[37] E. Tseng, F. Yu, Y. Yang, F. Mannan, K. St. Arnaud, D. Nowrouzezahrai, J.-F. Lalonde and F. Heide, "Hyperparameter optimization in black-box image processing using differentiable proxies," *ACM Transactions on Graphics,* vol. 38, no. 4, p. 1–14, 2019.