

DISS. ETH NO. 27078

Reconstruct Multi-body Dynamics for Compositional Re-simulation

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Shengyu Huang

MSc in Geomatik, ETH Zurich

born on 17.11.1995

citizen of China

accepted on the recommendation of

Prof. Dr. Konrad Schindler, examiner

Prof. Dr. Or Litany, co-examiner

Prof. Dr. Angela Dai, co-examiner

Prof. Dr. Simon Lucey, co-examiner

Prof. Dr. Andreas Wieser, co-examiner

2024

Abstract

placeholder for abstract

Kurzfassung

german text

german text

Acknowledgements

Thank you

Konrad, Andreas, Zan, Iro, Or, Sanja, Federico Michael, Michael, Keisuke, Titas, Lukas

Jiahui, Zian

Mikhail, Anton, Nico, Nikolai, Riccardo, Rodrigo, Alex, Nando, Ozgur, Manu, Corrine, Stefano, Bingxin, Binbin, Yujia, Torben,

Liyuan, Yuru, Hanfeng, Tao, Yuanwen,

Jixuan, Ye, Jimeng, Yikui, Lukas.

Lei Ke

Contents

Abstract	iii
Kurzfassung	v
Acknowledgements	vii
1. Introduction	1
1.1. Outline & Contributions	1
1.1.1. Predator	1
1.2. Relevance to Science and Economy	2
2. Background	3
2.1. Point cloud register	3
3. Neural LiDAR Fields for Novel View Synthesis	5
3.1. Introduction	6
3.2. Related Work	7
3.3. Background	8
3.3.1. Volume rendering for passive sensors	9
3.3.2. LiDAR model	9
3.4. LiDAR Novel View Synthesis	11
3.4.1. Neural scene representation	12
3.4.2. Volume rendering for LiDAR rays	12
3.4.3. Assembling the beam from multiple rays	13
3.4.4. Training the neural LiDAR field	14
3.5. Experiments	15
3.5.1. Datasets and Evaluation setting	15
3.5.2. Evaluation of LiDAR novel view synthesis	17
3.5.3. Downstream evaluation of novel views	19
3.6. Limitations and future work	20
3.7. Appendix	21
3.7.1. Datasets and implementation details	21
3.7.2. Implementation details	21
3.7.3. Methodology and loss functions	23
3.7.4. Additional results	23
4. Conclusions	35
4.1. Lessons Learned	35
4.2. Limitations	35

Contents

4.3. Outlook	35
A. Bibliography	37
B. List of Publications	43
C. Curriculum Vitae	45

1 | Introduction

paper dissertations require an introduction which includes the overriding research question, the methodology used and the relevance of the thesis to science and economy the conclusion should merge the results of the publications and include suggestions for ongoing research

1.1. Outline & Contributions

The work presented in this thesis comprises articles published to journals and conferences of the fields of Computer Vision and Experimental Fluid Dynamics.

1.1.1. Predator

In Chapter ...

Contributions.

- A warping-based variational optical flow approach is adapted to volumetric particle flow estimation. To account also for larger displacements between time steps, a coarse-to-fine pyramid scheme is implemented. An efficient primal-dual optimization approach is used to minimize the energy, which is formulated as a saddle-point problem.
- The energy formulation is derived from the stationary Stokes equations, which model the incompressibility and viscosity of the fluid. Both, a hard constraint and an alternative soft constraint variant on the flow divergence are integrated into the variational framework. The Euler-Lagrange equations of the energy reveal that the viscosity of the fluid can be accounted for via quadratic regularization of the flow gradient.
- To handle also large measurement volumes, a semi-dense formulation is introduced. It takes into account the individual requirements of particle reconstruction and flow estimation. While the data term is evaluated at full resolution, the flow field is regularized at a lower spatial resolution.
- Extensive evaluations are carried out for different data terms and regularizers. Further, ablation studies on the matching window size, seeding density and stepsize of the semi-dense formulation are performed, justifying the chosen parameters and the effectiveness of the presented approach.

1. Introduction

1.2. Relevance to Science and Economy

xxx

Open Source.. xxx

Scientific Recognition.. xxx

2 | Background

placeholder

2.1. Point cloud register

3 | Neural LiDAR Fields for Novel View Synthesis

Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, Or Litany

IEEE International Conference on Computer Vision, 2023

(Author version; for typeset version please refer to the original conference paper.)

Abstract

We present Neural Fields for LiDAR (NFL), a method to optimise a neural field scene representation from LiDAR measurements, with the goal of synthesizing realistic LiDAR scans from novel viewpoints. NFL combines the rendering power of neural fields with a detailed, physically motivated model of the LiDAR sensing process, thus enabling it to accurately reproduce key sensor behaviors like beam divergence, secondary returns, and ray dropping. We evaluate NFL on synthetic and real LiDAR scans and show that it outperforms explicit reconstruct-then-simulate methods as well as other NeRF-style methods on LiDAR novel view synthesis task. Moreover, we show that the improved realism of the synthesized views narrows the domain gap to real scans and translates to better registration and semantic segmentation performance.

[Project page]

3. Neural LiDAR Fields for Novel View Synthesis

3.1. Introduction

The goal of novel view synthesis is to generate a view of a 3D scene, from a viewpoint at which no real sensor image has been captured. This offers the possibility to observe *real* scenes from a *virtual*, unobserved perspective. Among other applications, it has tremendous potential for autonomous driving: synthetic novel views may be used to train and test perception algorithms across a wider range of viewing conditions, thus enhancing robustness and generalization. Moreover, novel view synthesis becomes critical when the desired viewpoints are not known in advance, *e.g.*, during training of a planning module whose decisions determine future vehicle locations.

Neural radiance fields (NeRFs) have led to unprecedented visual quality when synthesizing novel camera views [33, 2, 61, 34]. These methods represent the 3D scene in form of continuous density and radiance fields, from which images can be generated through volume rendering, mimicking the image acquisition process. The inductive bias of neural networks imparts NeRFs the ability to interpolate complex lighting and reflectance behaviours with a high degree of realism.

While most prior works focused on synthesizing camera views, 3D perception in the autonomous driving context typically relies partly (or even exclusively) on LiDAR measurements. Synthesizing realistic LiDAR scans from novel viewpoints thus has a lot of potential for data augmentation and closed-loop testing of autonomous navigation systems.

The problem of synthesizing novel LiDAR views has previously been addressed in two stages [28]. First, extract an explicit surface representation such as surfels or a triangular mesh from the scanned point clouds. Then, simulate LiDAR measurements from a novel viewpoint by casting rays and intersecting them with the surface model. Like for images, explicit reconstruction (which is not optimised towards the subsequent synthesis step) suffers from discretization artifacts and introduces noticeable errors [53]. Moreover, the rendering assumes an idealised ray model and neglects the divergence of the LiDAR beams, which causes frequent second returns from distant surfaces.

Here, we instead build on a main insight of NeRF [33]: directly optimizing an implicit scene representation for novel view synthesis can produce more realistic outputs than the reconstruct-then-simulate approach. Specifically, we propose Neural Fields for LiDAR (NFL), a NeRF-style representation for synthesizing novel LiDAR viewpoints.

Several NeRF extensions have utilized range measurements as additional supervision, and have shown that constraining the scene geometry more tightly can yield better (camera) view synthesis [9, 43]. Yet, the output of those methods are synthetic images, not LiDAR scans, consequently they have not paid attention to effects specific to LiDAR sensing: a laser scanner does *not* directly sense range, rather it measures the returned light energy per ray and determines the range based on the waveform. This includes the possibilities that there are multiple returns¹ from the emitted ray, or no return at all.

Our formulation closely adheres to the principles of the LiDAR measurement process and incorporates them into the neural field framework. Specifically, we (i) **devise volume**

¹In principle there can be >2 returns, but automotive LiDAR sensors typically record the first two echos.

rendering for LiDAR sensors; (ii) **incorporate beam divergence** and (iii) **propose truncated volume rendering** to account for secondary returns and improve range prediction.

We evaluate our method on both synthetic and real LiDAR data. To this end, we (iv) **develop a LiDAR simulator** for synthesizing scenes from 3D assets that serve as a test bed for viewpoints far from the original scan locations, and to study the effect of different scan patterns. Real data from the Waymo [48] dataset is used to evaluate NFL against real scans at held-out viewpoints, including real-world intensities, ray drops and secondary returns. Additionally, we (v) **propose a novel closed-loop evaluation protocol** that leverages real data to evaluate view synthesis in challenging views. As an end-to-end test for downstream tasks, we further evaluate the performance of state-of-the-art segmentation and registration networks when trained on real scans and tested on novel views generated by NFL.

3.2. Related Work

LiDAR simulation. Simulating realistic LiDAR data is useful for training perception models. Different from real-world LiDAR data that requires annotation efforts, simulated data can be automatically generated with ground truth labels, *e.g.* object bounding boxes and semantic segmentation. Unrealistic LiDAR simulation will prevent the trained models from generalizing to real data. Traditional simulation engines, such as those proposed in [10, 22], require the specification of sensor parameters and 3D scene assets and use ray-casting methods for simulation. Although these point clouds can accurately represent scene geometry, they often exhibit a discrepancy, or "domain gap", compared to real data, due to the lack of modeling for sensor noise, such as ray drop and Gaussian beam. Furthermore, this approach relies heavily on the creation of 3D scene assets, which can be time-consuming and expensive. To address these challenges, LiDARsim [28] reconstructs the static and dynamic scene assets from real data using surfel [39] representation and models the ray-drop pattern for improved realism. BaiduSim [11] proposes a probability map to model scene compositions in order to reduce the domain gap. Most recent work [13] learns to enhance existing simulated LiDAR intensity and ray-drop patterns, using the available corresponding RGB images.

Weather conditions, such as fog or rain, can significantly impact the quality of LiDAR data, and downstream models trained solely on ideal weather conditions may fail to generalize to these effects. Recent methods and datasets [15, 14, 19, 6] have been proposed to address this issue. SnowSIM [14] and FogSIM [15] sample snow particles and model the impulse response from atmospheric attenuation, respectively, to alter the range measurements of each ray. Other approaches [47, 19, 23] simulate LiDAR data on rainy days and the spray effects, in a similar fashion.

NeRF for Novel View Synthesis. NeRF [33] maps 5D position and direction to density and radiance scene values, and uses volume rendering [29, 30] to estimate pixel color. This technique has proven effective for generating realistic images at unseen camera views. Many methods have been proposed to improve robustness to camera poses [25, 8], handle dynamics [37, 40], anti-alias [62, 2, 3], and speed up optimisation [26, 61, 34] *etc.* Despite its high-quality novel view synthesis capacity, the underlying geometry of NeRF is considered inaccurate.

3. Neural LiDAR Fields for Novel View Synthesis

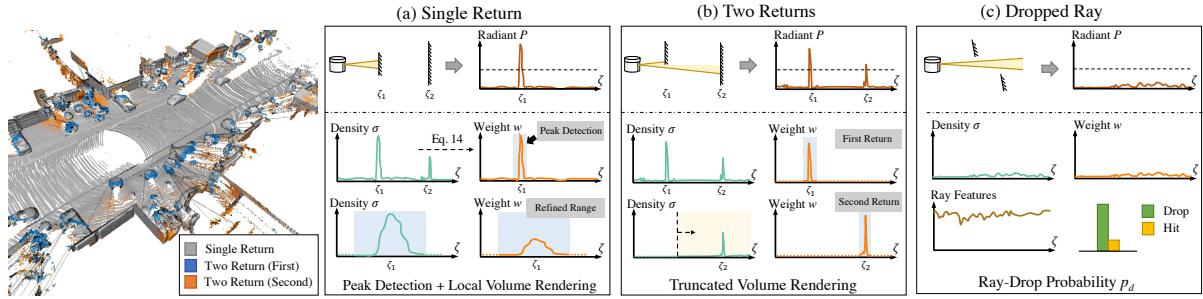


Figure 3.1.: Left: real LiDAR scan demonstrating key LiDAR return properties: a single return and two returns (first return shown in blue and second return in orange). Right: NFL models the waveform and accurately reproduces these properties. (a) Top: the LiDAR energy is fully scattered by the first surface. Bottom: NFL estimates range via peak detection on the computed weights w followed by volume rendering based range refinement. (b) Top: secondary returns resulting from a beam hitting two surfaces. Bottom: NFL employs beam divergence and a truncated volume rendering to estimate the second return. (c) Top: beams that do not hit a surface do not return detectable signal. Bottom: NFL utilizes geometric and semantic features to predict the ray drop probability. Refer to section 3.4.3 for more details.

rate and noisy [35], making it less favoured for geometry reconstruction, especially in sparse-views settings. [35, 60, 55] address this challenge by using implicit surface representations, and defining the density functions based on them to enabling volume rendering. DS-NeRF [9] and DenseDS-NeRF [44] use sparse depth supervision from SfM [46] points to regularise the density field. Urban Radiance Field [43] leverages LiDAR data for depth supervision.

Neural fields beyond regular cameras.. Neural fields are a natural and continuous representation [58] for spatio-temporal information including SDFs [38], occupancy [31] and radiance field [33] etc. While in its original form, NeRF [33] performs novel view synthesis using tonemapped low dynamic range images, RawNeRF [32] extends it to operate over the high dynamic range images, enabling additional adjustments to focus, exposure, and tonemapping. Törf [1] incorporates the image formation model for continuous-wave Time-of-Flight (ToF) cameras into NeRF, allowing it to jointly process RGB and ToF sensor data and improve reconstruction robustness to large motions. EventNeRF [45] and ENeRF [21] optimise the scene representation for Novel View Synthesis (NVS) from sparse event streams that contain asynchronous per-pixel brightness change signals. Other works [41, 27] explore the use of acoustic signals for surface reconstruction or NVS.

3.3. Background

We start by reviewing the principles of volume rendering (§ 3.3.1) and the sensor model for LiDAR (§ 3.3.2). This sets the stage for the proposed formulation of Neural LiDAR Fields (Sec. 3.4).

3.3.1. Volume rendering for passive sensors

In the following, we provide a brief summary of camera-based volume rendering as used by NeRF [33, 49]. This will serve as the basis to derive volume rendering equations for the active LiDAR sensor.

Density and transmittance.. For a ray $\mathbf{r}(\mathbf{o}, \mathbf{d})$ emitted from the origin $\mathbf{o} \in \mathbb{R}^3$ in direction $\mathbf{d} \in \mathbb{R}^3$, the *density* σ_ζ at range ζ is a scalar function that indicates the differential likelihood of hitting a reflective particle at position $\mathbf{r}_\zeta = \mathbf{o} + \zeta \mathbf{d}$. *Transmittance* T_ζ indicates the probability of traversing the interval $[0, \zeta)$ without hitting anything. Taking a differential step $d\zeta$ along the ray, the probability of *not* hitting anything is $T_{\zeta+d\zeta} = T_\zeta \cdot (1 - \sigma_\zeta d\zeta)$. Integrating over an interval $[\zeta_0, \zeta]$ yields the probability $T_{\zeta_0 \rightarrow \zeta}$ of traversing the interval unhindered,

$$T_{\zeta_0 \rightarrow \zeta} \equiv \frac{T_\zeta}{T_{\zeta_0}} = \exp\left(-\int_{\zeta_0}^{\zeta} \sigma_t dt\right), \quad (3.1)$$

leading to the decomposition: $T_\zeta = T_{0 \rightarrow \zeta_0} \cdot T_{\zeta_0 \rightarrow \zeta}$.

Integration over homogeneous media.. Assuming a homogeneous medium along the ray segment $[\zeta_j, \zeta_{j+1}]$ with constant radiance $\mathbf{c} \in \mathbb{R}^3$ and density σ , the accumulated radiance from that segment evaluates to

$$\mathbf{c}(\zeta_j \rightarrow \zeta_{j+1}) = \mathbf{c}_{\zeta_j} \int_{\zeta_j}^{\zeta_{j+1}} T_{\zeta_j \rightarrow \zeta} \cdot \sigma_\zeta d\zeta = \alpha_{\zeta_j} \mathbf{c}_{\zeta_j}, \quad (3.2)$$

with $\alpha_{\zeta_j} = 1 - \exp(-\sigma_{\zeta_j}(\zeta_{j+1} - \zeta_j))$ being the *opacity*.

Volume rendering.. By discretizing the ray into N segments with piecewise constant densities and radiance values, we obtain the total irradiance (color to be rendered):

$$\mathbf{c} = \sum_{j=1}^N \int_{\zeta_j}^{\zeta_{j+1}} T_\zeta \cdot \sigma_\zeta \mathbf{c}_\zeta d\zeta = \sum_{j=1}^N w_j \mathbf{c}_{\zeta_j}, \quad (3.3)$$

where w_j is the *weight* for the j -th segment:

$$w_j = \alpha_{\zeta_j} \prod_{k=1}^{j-1} (1 - \alpha_{\zeta_k}). \quad (3.4)$$

3.3.2. LiDAR model

LiDAR emits laser beam pulses and determines the distance from the sensor to the nearest reflective surface by measuring the time of flight. Often the LiDAR beams are pictured as ideal straight-line segments ending on a 3D surface point. In reality, things are more complicated: real lasers emit a pulse with non-zero divergence and finite pulse width, while real receivers

3. Neural LiDAR Fields for Novel View Synthesis

employ signal processing techniques like radiant thresholding and binning to detect the return. This leads to phenomena such as discretization errors, over- and underestimation biases (*c.f.* Fig. 3.2), and multiple returns from one beam (or no return at all). In the following, we discuss key aspects of the LiDAR acquisition process and explain the effects that emerge, which inspire our model design. We also built a LiDAR simulator that accounts for these mechanisms, see § 3.5.1.

Beam divergence.. LiDAR beams diverge as they travel away from the sensor. The size of laser beams can become wider over distance, and typically not negligible in street scenes. Consequently, the illuminated area grows and the irradiance (radian power per area) decreases with increasing range. The size of the beam’s footprint is characterised by the divergence angle ($2\gamma_0$) and the range ζ . Let \mathbf{r}^γ be an ideal ray within the beam’s cross-section, $\gamma \leq \gamma_0$, then its irradiance $E(\zeta, \gamma)$ at range ζ can be approximated by a Gaussian function in the ray coordinate system [54]:

$$E(\zeta, \gamma) = \frac{2I_0}{\pi(\gamma_0\zeta)^2}g(\gamma), \quad g(\gamma) = \exp\left(-2\frac{\gamma^2}{\gamma_0^2}\right), \quad (3.5)$$

where I_0 is the pulse peak power.

Pulse waveform.. When the emitted LiDAR pulse returns to the sensor, the range to the reflective surface can be determined from its travel time and the speed of light c . Since the pulse has finite duration τ_H , the time of return is found by analysing the received intensity profile. The transmitted pulse power over time can be characterised as [7]:

$$P_e(t) \propto \left(\frac{t}{\tau}\right)^2 \exp\left(-\frac{t}{\tau}\right), \quad \tau = \frac{\tau_H}{1.75}. \quad (3.6)$$

The range-dependent received radiant power $P(\zeta)$ is the result of convolving the pulse power with the systems impulse response $H(\zeta)$ [42, 15, 14]:

$$P(\zeta) = \int_0^{2\zeta/c} P_e(t)H(\zeta - \frac{ct}{2}) dt, \quad (3.7)$$

where the impulse response $H(\zeta)$ is a composition of the target and the receiver responses: $H(\zeta) = H_T(\zeta)H_C(\zeta)$. Assuming a Lambertian surface, the target response due to a surface located at range ζ_0 depends on the incidence angle θ and the reflectance ρ :

$$H_T(\zeta) = \frac{\rho}{\pi} \cos(\theta)\delta(\zeta - \zeta_0), \quad (3.8)$$

with $\delta(\cdot)$ the Dirac delta function. The receiver response $H_C(\zeta)$ is computed by integrating over the solid angle spanned by the receiver’s effective area A_e :

$$H_C(\zeta) = T_\zeta^2 \frac{A_e}{\zeta^2}, \quad (3.9)$$

where $T_\zeta \in [0, 1]$ is the one-way transmittance, squared to account for the two-way trip.

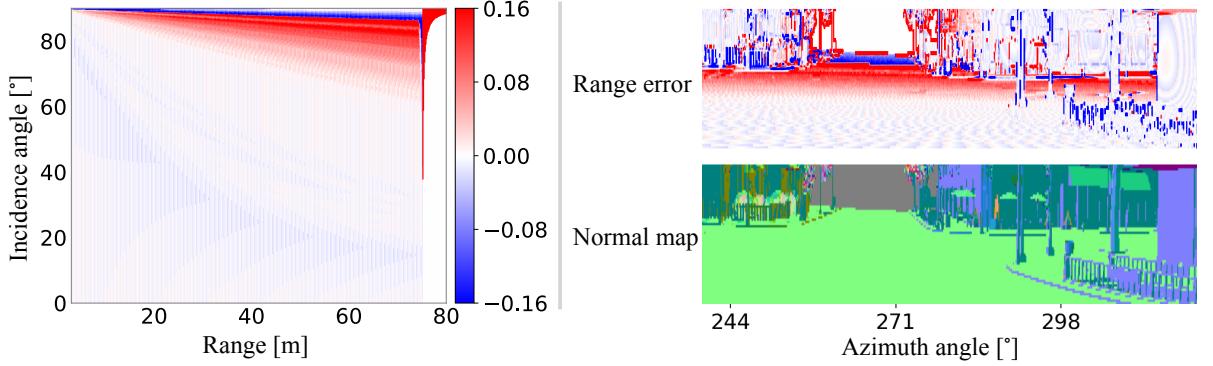


Figure 3.2.: The range accuracy of the LiDAR sensor is affected by waveform discretization and beam divergence. The LiDAR sensor has a tendency to overestimate range in high incidence angle regime, which becomes increasingly pronounced at higher range regimes (left). This is also reflected on *TownReal* dataset (right).

Beam discretization. In practice, we follow [57] and approximate the Gaussian beam profile using $M = 37$ rays that are radially distributed around the central ray with different divergence angles γ_i . The total radiant power $P(\zeta)$ is the weighted sum over those rays: $P(\zeta) = \sum_{i=1}^M g(\gamma_i)P_i(\zeta)$. Taking into account the beam divergence is important to reproduce two important phenomena: range biases and multiple returns, see Fig. 3.1 and Fig. 3.2. As different rays hit a slanted surface at different ranges the integrated waveform peak may shift, causing over- or underestimations. Along object edges, rays within the same beam may hit different surfaces, causing multiple peaks (respectively, range readings), in the return waveform.

Range estimation..

One common approach to estimate the surface range from the received waveform is to locate its peak. To that end the signal is discretized in time to obtain a histogram, and local maxima above a certain threshold are declared detections [57]. The associated range values are then corrected to remove known biases stemming from the pulse waveform (*c.f.* Eq. (3.6)) and, optionally, biases due to the radiant power [57]. By modeling the binning and thresholding procedure one can reproduce further LiDAR behaviors: systematic discretization errors in the range resolution (*c.f.* Fig. 3.2), and the dropping of rays with low returned power (*c.f.* Fig. 3.1).

3.4. LiDAR Novel View Synthesis

We now turn to constructing a neural field model tailored for LiDAR scans, along with a differentiable volume rendering scheme to enable LiDAR novel view synthesis. We first formulate the problem setting, then set up a corresponding neural scene representation (§ 3.4.1) and derive volume rendering for active sensing (§ 3.4.2). Finally, we describe the rendering procedure used to synthesize novel views (§ 3.4.3) and our optimisation scheme (§ 3.4.4).

Problem setting. Consider a collection of LiDAR scans $\mathcal{X} = \{\mathbf{X}_v\}_{v=1}^{n_v}$ captured by a moving sensor (*e.g.*, mounted on a vehicle). Each scan \mathbf{X}_v is associated with a sensor pose $\mathbf{T}_v \in \text{SE}(3)$

3. Neural LiDAR Fields for Novel View Synthesis

and consists of n_r rays. Every ray $\mathbf{r}(\mathbf{o}, \mathbf{d})$ records observations $(\zeta_1, e_1, p_d, p_s, \zeta_2, e_2)$: the range ζ_1 and intensity e_1 of the first return; a ray drop flag $p_d \in \{0, 1\}$; a two-return mask $p_s \in \{0, 1\}$; and range ζ_2 and intensity e_2 values of the second return. Our goal is to reconstruct a (continuous) volumetric representation of the scene in terms of density σ and reflectance ρ , from which we can subsequently render virtual LiDAR scans \mathbf{X}_{tgt} from novel sensor poses \mathbf{T}_{tgt} .

3.4.1. Neural scene representation

We encode the scene as a neural field $F : (\mathbf{x}, \mathbf{d}) \mapsto (\sigma, \rho, p_d)$ that takes as input a location $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^3$, and returns a density σ , a reflectance ρ and a ray drop probability p_d . We found it beneficial to additionally return also a local contribution $p_d \in [0, 1]$ to the probability of ray drop, which will be discussed below. Technically, we use a hash encoding [34] to map coordinates \mathbf{x} to positional features $\mathbf{f}_{pos} \in \mathbb{R}^{32}$ and project the view direction onto the first 16 coefficients of the spherical harmonics basis, $\mathbf{f}_{dir} \in \mathbb{R}^{16}$. The neural field is parameterized by four Multi-Layer Perceptrons (MLPs): $[\sigma; \mathbf{f}_{geo}] = f_\sigma(\mathbf{f}_{pos})$ regresses density and extracts an additional geometry feature $\mathbf{f}_{geo} \in \mathbb{R}^{15}$ that supports the other networks; $\rho = f_\rho(\mathbf{f}_{geo}, \mathbf{f}_{dir})$ regresses reflectance; $p_d = f_{drop}(\mathbf{f}_{geo}, \mathbf{f}_{dir})$ classifies whether a ray drop occurs; and $p_s = f_{sr}(\mathbf{f}_{beam})$ classifies the existence of a second return. The feature \mathbf{f}_{beam} will be detailed in § 3.4.3.

3.4.2. Volume rendering for LiDAR rays

In contrast to passive sensors like cameras that rely on ambient illumination, LiDAR actively illuminates the scene and measures the back-scattered radiance. This two-way transmittance alters the volume rendering formulation.

Radiant power integration. As discussed in § 3.3.2 the radiant power along a LiDAR ray is a delta function that is non-zero only at reflecting surfaces. To incorporate this forward model into the volumetric representation we combine Eq. (3.8) and Eq. (3.9) to obtain the probabilistic radiant power:

$$P_\zeta = C \frac{T_\zeta^2 \cdot \sigma_\zeta \rho_\zeta}{\zeta^2} \cos(\theta), \quad (3.10)$$

where C is a system constant, ρ_ζ is the differentiable reflectance, and θ is the incidence angle. In a homogeneous medium with constant reflectance ρ and density σ , the integrated $P(\zeta_j \rightarrow \zeta_{j+1})$ evaluates to:

$$P(\zeta_j \rightarrow \zeta_{j+1}) = \int_{\zeta_j}^{\zeta_{j+1}} C \frac{T_{\zeta_j \rightarrow \zeta}^2 \sigma \rho_\zeta}{\zeta^2} \cos(\theta_j) d\zeta \approx \alpha_{\zeta_j} \rho'_{\zeta_j}, \quad (3.11)$$

where we approximate $\zeta \in [\zeta_j, \zeta_{j+1}]$ by $\frac{\zeta_j + \zeta_{j+1}}{2}$, and

$$\alpha_{\zeta_j} = \frac{1}{2} \left(1 - e^{-2\sigma_{\zeta_j} \delta_j} \right), \quad \rho'_{\zeta_j} = C \rho_{\zeta_j} \frac{4 \cos(\theta_j)}{(\zeta_j + \zeta_{j+1})^2}. \quad (3.12)$$

Volume rendering. The observed power at the active sensor can be evaluated by plugging Eq. (3.11) into Eq. (3.3):

$$P = \sum_{j=1}^N \int_{\zeta_j}^{\zeta_{j+1}} C \frac{T_\zeta^2 \cdot \sigma_\zeta \rho_\zeta}{\zeta^2} \cos(\theta_j) d\zeta = \sum_{j=1}^N w_j \rho'_{\zeta_j}, \quad (3.13)$$

where the weights w_j are now evaluated as (*c.f.* Eq. (3.4)):

$$w_j = 2\alpha_{\zeta_j} \cdot \prod_{k=1}^{j-1} (1 - 2\alpha_{\zeta_k}). \quad (3.14)$$

3.4.3. Assembling the beam from multiple rays

Next, we apply the adapted volume rendering formulation to multiple rays within a single LiDAR beam.

First range estimation. We adopt a two-stage approach to extract range values from the neural field,² as shown in Fig. 3.1 (a). To estimate the range for an ideal ray \mathbf{r} , we uniformly sample N^c points and query their density values, then compute the weights $\{w_j^c\}_{j=1}^{N^c}$ using Eq. (3.14). A coarse peak estimate ζ_p is obtained by finding the point with the highest weight along the ray: $p = \text{argmax}_j \{w_j^c\}_{j=1}^{N^c}$. Next, we uniformly sample N^f points from the local interval $\zeta_j \in [\zeta_p - \epsilon, \zeta_p + \epsilon]$. The weights w_j^f at these points are recomputed and normalized to then obtain the final, refined range estimate ζ_f as: $\zeta_f = \sum_{j=1}^{N^f} w_j^f \cdot \zeta_j$.

Second range estimation. As discussed in § 3.3.2 a single LiDAR beam might have multiple returns if enough energy was reflected from surfaces further away than the first return. To capture this behavior in our scene representation, we employ *truncated* volume rendering to estimate the radiant power beyond the first return (see Fig. 3.1 (b)).

Specifically, for each beam, we first predict a two-return mask p_s , by classifying its features $\mathbf{f}_{\text{beam}} = (\bar{\mathbf{f}}_{\text{geo}}, \mathbf{f}_{\text{dir}}, \mathbf{f}_{\text{range}})$, where $\bar{\mathbf{f}}_{\text{geo}}$ is the volume-rendered geometric feature, and $\mathbf{f}_{\text{range}}$ describes the standard deviation and maximum discrepancy of range estimates at the first return. Intuitively, $\bar{\mathbf{f}}_{\text{geo}}$ describes the local geometry (*e.g.* an edge), \mathbf{f}_{dir} encodes the relation of the beam to the geometry, and $\mathbf{f}_{\text{range}}$ characterizes the beam’s prior interaction with the scene.

For beams that have two returns, we then perform *truncated* volume rendering as follows. We first add a buffer ξ ³ to the estimated range ζ_1 of the first return. We then reset the transmittance $T_{\zeta_1+\xi}$ to 1 by zeroing out the densities up to $(\zeta_1 + \xi)$ and recalculate the weights to ensure that they adhere to Eq. (3.14). Finally, we repeat the range estimation described above to estimate the range of the second return ζ_2 . Note that for beams with two returns, the estimated range ζ_1 denotes the minimum range of all rays within the beam diameter, i.e. we perform vol-

²Note the similarity to the detector in the instrument that first finds the peak of the waveform, then corrects for pulse shape.

³The buffer ξ is sensor specific and describes the minimum spacing between two distinct returns.

3. Neural LiDAR Fields for Novel View Synthesis

ume rendering on all rays of a beam and pick the closest one as the first return. This is different from the beams with a single return where we directly use the central ray to estimate ζ_1 .

Reflectance estimation. At every detected surface point we can also retrieve reflectance from the neural field, using the relation $\rho = \sum_{j=1}^{N^f} w_j^f \cdot \rho_j$.

Ray drop probability. In real LiDAR sensors, some emitted beams return no range measurement at all. This happens when the observed return signal has either too low amplitude or no clear peak (Fig. 3.1 (c)). However, this effect is hard to model in a fully physics-based way,⁴ because it depends on (usually undisclosed) details of the detection logic. We empirically observe that the ray drop probability can be learned from LiDAR measurements. To this end, we augment the neural scene representation with a dedicated variable for the local probability of *not* back-scattering radiant power $p_d(\zeta) \in \{0, 1\}$ ⁵. Volume rendering integrates that quantity into a ray drop probability: $p_d(\mathbf{r}) = \sum_{j=1}^{N_c} w_j^c \cdot p_d(\zeta_j)$.

3.4.4. Training the neural LiDAR field

Given a set of posed LiDAR scans, we optimise our neural field model by minimising the loss

$$\mathcal{L} = \mathcal{L}_{\text{range}} + \lambda_e \mathcal{L}_e + \lambda_d \mathcal{L}_d + \lambda_s \mathcal{L}_s , \quad (3.15)$$

consisting of a reconstruction loss $\mathcal{L}_{\text{range}}$ for range estimation, reflectance loss \mathcal{L}_e , and classification losses \mathcal{L}_d for ray drops and \mathcal{L}_s for two returns.

Range reconstruction. We add two separate losses for the coarse range ζ_p and the refined range ζ_f , $\mathcal{L}_{\text{range}} = \mathcal{L}_{\text{range}}^c + \mathcal{L}_{\text{range}}^f$. For coarse range, we impose a Gaussian distribution [43] around the ground truth $\hat{\zeta}$,

$$\mathcal{L}_{\text{range}}^c = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left(1 - \sum_{w_j \in \mathcal{X}_c^n} w_j \hat{w}_j + \sum_{w_k \in \mathcal{X}_c^e} w_k^2 \right) , \quad (3.16)$$

where \mathcal{R} is the set of LiDAR rays, \mathcal{X}_c^n and \mathcal{X}_c^e denote points sampled within and outside the interval $[\hat{\zeta} - \epsilon, \hat{\zeta} + \epsilon]$. The ground truth weight \hat{w}_j is calculated by integrating the Gaussian distribution. The range refinement loss is defined as: $\mathcal{L}_{\text{range}}^f = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} |\hat{\zeta} - \zeta_f|$.

Reflectance reconstruction. is optimized by minimizing an L2 loss w.r.t. the ground truth intensity \hat{e} : $\mathcal{L}_e = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} (\hat{e} - e)^2$.

Ray drop and dual return masks. are trained as classification tasks, by minimizing the com-

⁴Beyond simple thresholding, which our beam model would support.

⁵Please refer to the supplementary for discussions on this design choice.

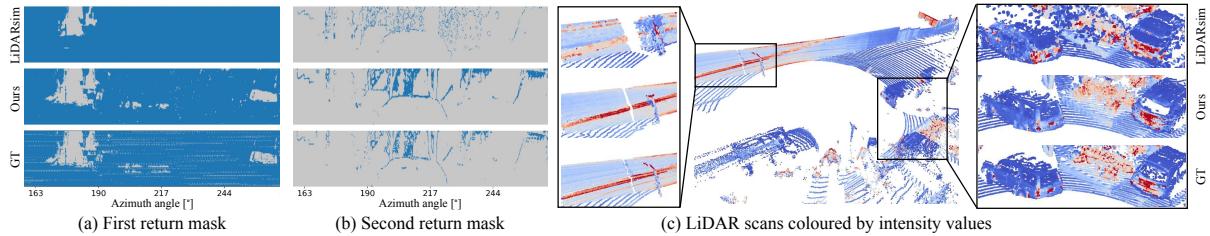


Figure 3.3.: Qualitative results of LiDAR novel view synthesis on *Waymo Interp.* dataset. On the left, we color-code rays **with** and **without** return. On the right side, LiDAR intensity values are color-coded as :0 0.25.

Method	First range			Second range				Intensity		Ray drop			
	Recall@50↑	MAE↓	MedAE↓	Seg. recall↑	Seg. precision↑	Recall@50↑	MAE↓	MedAE↓	MSE ^{1st} ↓	MSE ^{2nd} ↓	Recall↑	Precision↑	IoU↑
LiDARsim [28]	74.1	105.4	18.5	3.5	11.5	1.0	2258.0	1898.2	0.013	0.018	32.5	85.5	30.5
Central ray	92.8	32.8	5.6	79.8	62.9	61.1	589.1	21.8	0.004	0.009	64.3	81.7	57.1
Ours	92.3	36.1	5.7	82.1	55.6	67.4	505.1	13.4	0.004	0.008	65.1	78.0	56.1
GT mask	93.2	29.7	5.6	100.0	100.0	79.8	116.0	8.1	0.004	0.011	65.1	78.0	56.1

Table 3.1.: Comprehensive ray measurement evaluation of LiDAR novel view synthesis on *Waymo Interp.* dataset.

bination of a binary cross entropy loss \mathcal{L}_{bce} and a Lovasz loss \mathcal{L}_{ls} [5]:

$$\mathcal{L}_* = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} (\mathcal{L}_{bce}(p_*, \hat{p}_*) + \mathcal{L}_{ls}(p_*, \hat{p}_*)) . \quad (3.17)$$

3.5. Experiments

We start by describing our LiDAR simulator, datasets, evaluation metrics, and baselines in § 3.5.1. In § 3.5.2, we evaluate NFL directly on the LiDAR novel view synthesis task. Finally, in § 3.5.3 we evaluate the suitability of our synthesized LiDAR data for two low-level tasks, point cloud registration and semantic segmentation.

3.5.1. Datasets and Evaluation setting

LiDAR simulator – TownReal dataset. To enable quantitative evaluation in a controlled environment, we build a LiDAR simulator that allows us to virtually scan synthetic 3D assets represented either as triangular meshes or surfels. Specifically, we follow the LiDAR model described in § 3.3.2 and allow control over the angular resolution, beam divergence, and pulse shape of the LiDAR sensor.

We use this simulator in combination with a 3D asset of a town [18] to synthesize the *Town* dataset. We generate four scenes by splitting the 3D asset into four non-overlapping areas. Training and test scans are created from different trajectories. We use two different configurations of the LiDAR sensor: (*i*) *TownClean*, in which LiDAR scans are simulated using an idealized, non-divergent ray; and (*ii*) *TownReal*, with a diverged beam profile approximated

3. Neural LiDAR Fields for Novel View Synthesis

Method	TownClean			TownReal			Waymo interp.			Waymo NVS		
	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓
i-NGP [34]	42.2	4.1	17.4	49.8	4.8	19.9	26.4	5.5	11.6	30.4	7.3	<u>15.3</u>
DS-NeRF [9]	<u>41.7</u>	3.9	<u>16.6</u>	<u>48.9</u>	4.4	<u>18.8</u>	<u>28.2</u>	6.3	14.5	30.4	<u>7.2</u>	16.8
URF [43]	43.3	4.2	16.8	52.1	5.1	20.7	28.2	<u>5.4</u>	12.9	43.1	10.0	21.2
LiDARsim [28]	159.6	0.8	23.5	162.8	<u>3.8</u>	27.4	116.3	15.2	27.6	160.2	16.2	34.7
Ours	32.0	<u>2.3</u>	9.0	39.2	3.0	11.5	30.8	5.1	<u>12.1</u>	<u>32.6</u>	5.5	13.2

Table 3.2.: Results of LiDAR novel view synthesis for the first range.

Method	TownClean			Waymo Interp.		
	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓
i-NGP [34]	41.0 (<u>-1.2</u>)	4.1 (<u>0.0</u>)	17.6 (<u>0.2</u>)	25.3 (<u>-1.1</u>)	4.5 (<u>-1.0</u>)	10.5 (<u>-1.1</u>)
DS-NeRF [9]	37.4 (<u>-4.2</u>)	3.0 (<u>-0.9</u>)	14.4 (<u>-2.2</u>)	27.4 (<u>-0.8</u>)	5.4 (<u>-1.0</u>)	13.6 (<u>-0.9</u>)
URF [43]	46.4 (<u>3.0</u>)	4.5 (<u>0.3</u>)	18.4 (<u>1.6</u>)	28.3 (<u>0.1</u>)	5.3 (<u>-0.1</u>)	13.1 (<u>0.2</u>)
Ours	32.0 (<u>-2.1</u>)	2.3 (<u>-2.5</u>)	9.0 (<u>-3.9</u>)	30.8 (<u>-2.1</u>)	5.1 (<u>-2.0</u>)	12.1 (<u>-2.3</u>)

Table 3.3.: Ablation study of volume rendering for active sensing.

via 37 subrays. See the supplementary material for further details.

Waymo Open dataset.. For evaluation on real-world data we use Waymo open dataset [48] which was captured by a 64-beam LiDAR sensor at 10 Hz. Here, we select four static scenes (see sequence IDs in supplementary material) and extract a five-second clip from each, resulting in 50 scans per scene. We hold out every 5-th frame as a test view and use the remaining 40 scans for training (*Waymo Interp.*)

To evaluate the methods in a more challenging setting we propose a novel evaluation protocol based on a closed-loop simulation (*Waymo NVS*). The protocol involves training and testing on all scans of a scene by first optimizing on the input views to synthesize novel views from a changed trajectory (shift the sensor by [1.5, 1.5, 0.5] meters⁶). The novel view are then used to re-optimize the method, synthesize scans in the original view and compare to the original scans to gauge performance. This formulation allows us to control task difficulty and could also be applied to evaluate camera-based novel view synthesis methods.

Evaluation metrics.. To evaluate range accuracy, we report four metrics: mean and median absolute errors (*MAE* [cm], *MedAE* [cm]), two-way Chamfer distance (*CD* [cm]), and *recall@50*, which denotes the percentage of rays with range errors below 50 cm. We additionally measure the two return segmentation recall (*Seg. recall*) and precision (*Seg. precision*). Intensity is evaluated using mean squared error (*MSE*). For ray drop segmentation, we report recall and precision [%], and intersection-over-union (*IoU* [%]). For point cloud registration, we report rotation error (*RE* [°]) and translation error (*TE* [cm]).

Baselines.. We compare NFL to four baselines. Closest to our problem setup is LiDARsim [28] which was designed for LiDAR synthesis based on surface reconstruction and ray-surfel cast-

⁶Please refer to the supplementary for ablations on different sensor shift configurations.

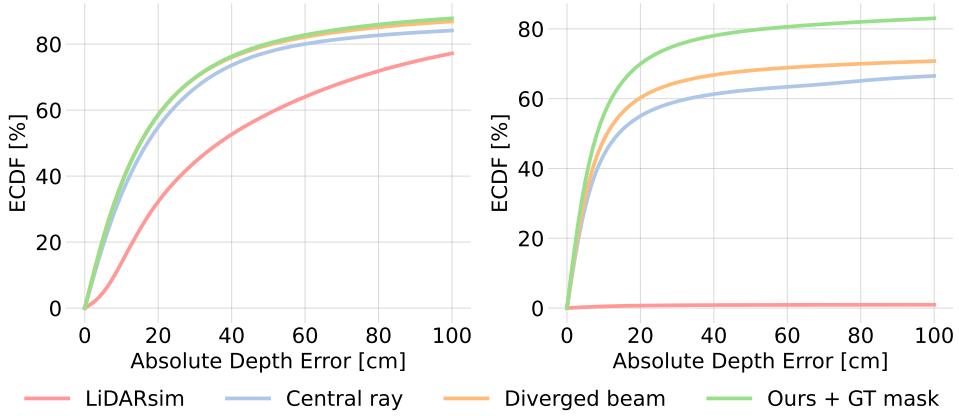


Figure 3.4.: Beam divergence modeling improves range accuracy of rays with dual returns. This is evident in the improved error distribution of the first (left) and second return range (right).

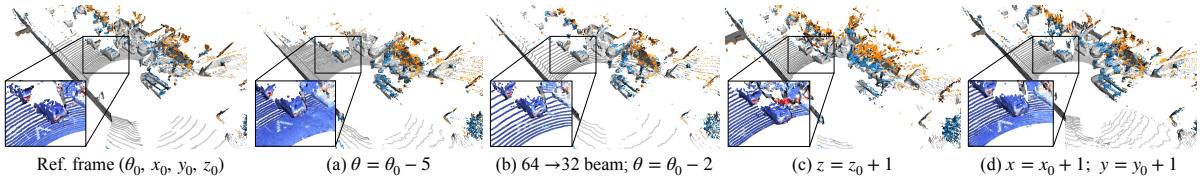


Figure 3.5.: LiDAR novel view synthesis by changing the sensor elevation angle $\theta[\circ]$, pose $(x, y, z)[m]$ and number of beams. Zoom-in points are color-coded by intensity values.

ing. We re-implement LiDARsim and augment it with a diverged beam profile to enable synthesis of the second returns. Additionally, we adapt three NeRF-like methods that were originally proposed for image synthesis i-NGP [34], DS-NeRF [9], and URF [43] by modifying their volumetric rendering to improve their range predictions. Additional details are available in the supplementary.

3.5.2. Evaluation of LiDAR novel view synthesis

Ray measurement..

Using the *Waymo Interp.* dataset we conduct a comprehensive analysis of all the ray measurements and present the results in Tab. 3.1 and Fig. 3.3. Only NFL and LiDARsim [28] are used for this experiment, as other baselines can only support a single return. LiDARsim’s surface representation is explicit and not optimized for novel view synthesis nor accounts for view-dependent effects, which results in inferior range prediction and difficulties in retrieving secondary returns. In contrast, NFL directly optimizes the neural field for view synthesis while accounting for LiDAR acquisition process characteristics, resulting in significantly reduced range errors and superior performance in intensity and ray drop probability estimation. Notably, equipping our model with a *diverged beam* representation improves range estimation for both first and second returns for rays with dual returns(*c.f.* Fig. 3.4). However, diverged beam does slightly degrade the overall first-range accuracy likely due to imprecise two-return mask

3. Neural LiDAR Fields for Novel View Synthesis

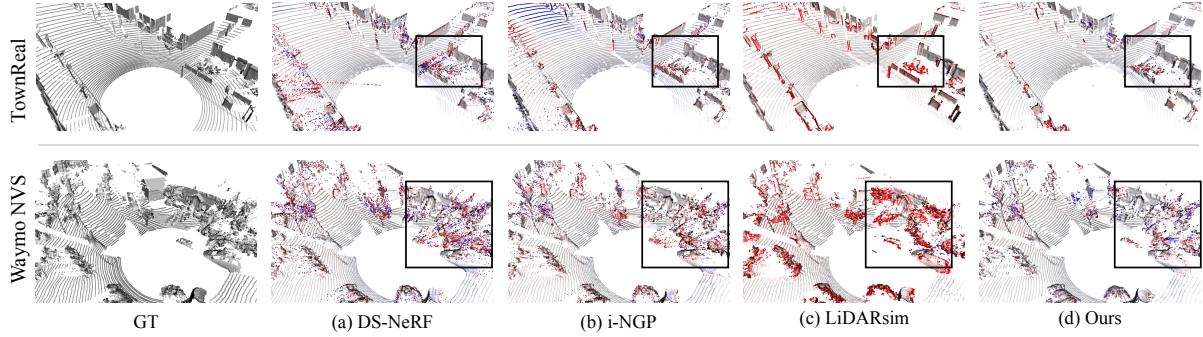


Figure 3.6.: Qualitative comparison of first range estimation. Regions with gross errors (-100 cm to 100 cm) are highlighted.

Method	TownClean			TownReal			Waymo NVS		
	Rec@5 \uparrow	RE \downarrow	TE \downarrow	Rec@5 \uparrow	RE \downarrow	TE \downarrow	Rec@2 \uparrow	RE \downarrow	TE \downarrow
i-NGP [34]	70.3	0.1	4.2	76.0	0.1	4.2	60.2	0.1	1.9
DS-NeRF [9]	58.3	0.2	5.1	56.2	0.2	5.1	42.3	0.1	2.4
URF [43]	61.5	0.2	5.0	59.9	0.1	4.7	32.1	0.1	2.7
LiDARsim [28]	82.8	0.1	3.4	<u>79.2</u>	0.1	3.4	<u>62.8</u>	0.1	<u>1.8</u>
Ours	<u>80.2</u>	0.1	<u>3.7</u>	85.9	0.1	3.4	71.9	0.1	1.7

Table 3.4.: Point cloud registration results on three datasets.

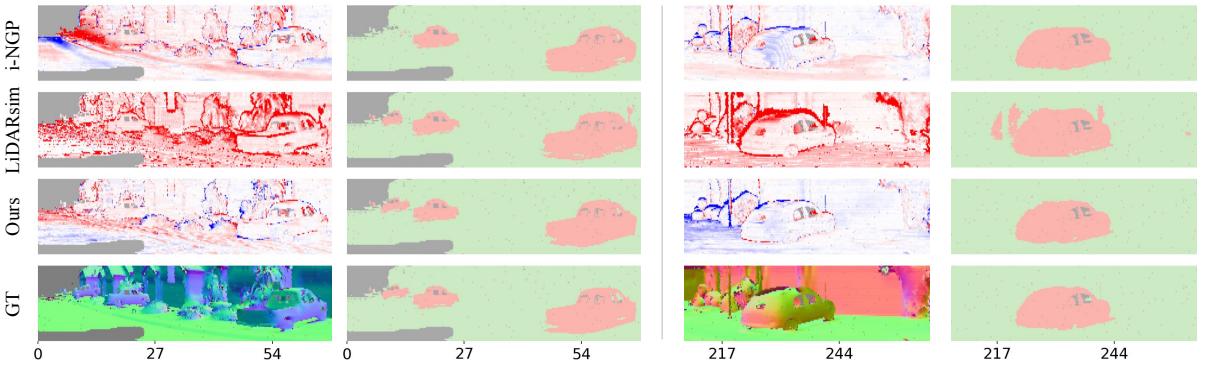
estimation. This hypothesis is supported by results using the ground truth two-return mask (*GT mask*). In Fig. 3.5 we show more qualitative results of novel view synthesis by NFL.

First range..

The results of estimating the range of the first return on all datasets are presented in Tab. 3.2 and Fig. 3.6. As demonstrated by the results on *TownClean*, *TownReal*, and *Waymo NVS*, the proposed volume rendering formulation of NFL effectively regularizes the density field resulting in superior performance in challenging cases. Even in the easier setting (resembling overfitting) on *Waymo Interp.* dataset, our method achieves competitive performance. In contrast, NeRF-like formulations (i-NGP [34], DS-NeRF [9], and URF [43]) perform poorly when evaluated on real novel views due to their inability to account for the active sensing principle. LiDARsim achieves promising results on datasets with simple geometry and clean LiDAR measurements, as evidenced by low *MedAE* scores on *TownClean* and *TownReal*. However, its explicit representation struggles with complex geometry in noisy real-world scenes, e.g., the vegetation regions in the *Waymo* dataset, resulting in high *MedAE* scores.

Ablation study of volume rendering for active sensing. To evaluate the effectiveness of our volume rendering formulation for active sensors, we replace the volume rendering [33] formulation initially developed for passive sensing in all NeRF-based baselines and report performance difference in Tab. 3.3. Our formulation improves range accuracy across all settings, without any hyper-parameter tuning.

Method	Vehicle			Background		
	Recall \uparrow	Precision \uparrow	IoU \uparrow	Recall \uparrow	Precision \uparrow	IoU \uparrow
i-NGP [34]	93.2	85.9	80.9	98.3	99.2	97.6
DS-NeRF [9]	90.7	87.1	80.2	98.5	98.9	97.4
URF [43]	87.8	81.7	73.7	98.0	98.4	96.5
Lidarsim [28]	90.5	70.5	65.9	94.9	99.0	94.0
Ours	95.9	87.0	83.9	98.3	99.5	97.8

Table 3.5.: Semantic segmentation results on *Waymo NVS* dataset.Figure 3.7.: Semantic segmentation results on synthesised *Waymo NVS* dataset. Geometry in-accuracy (-100 100 cm) leads to erroneous semantic segmentation (dropped rays, **vehicle**, **pedestrian**, **background**).

3.5.3. Downstream evaluation of novel views

Having demonstrated NFL’s improved ability to synthesize high-quality LiDAR scans through various metrics, we proceed to evaluate their perceptual quality by using them as input for two low-level perception tasks: point cloud registration [17] and semantic segmentation [51].

Point cloud registration. To evaluate the extent to which synthesized scans preserve local geometric features, we apply the same point cloud registration model [16] pre-trained on Waymo [48] to both GT LiDAR scans and scans synthesized using different methods. Tab. 3.4 shows that NFL outperforms the baseline methods on datasets with complex geometry and higher noise levels (*TownReal* and *Waymo NVS*) that are more susceptible to artifacts occurring as a result of the LiDAR acquisition process.

Semantic segmentation. To probe the potential domain gap between real and synthetic scans we apply the same, pre-trained semantic segmentation model [51] to both and compare the predictions. Tab. 3.5 depicts the performance for both the *vehicle* and *background* classes. Notably, NFL achieves the highest recall for the *vehicle* class, which is strongly affected by dual returns and ray drops. Example predictions are shown in Fig. 3.7.

3.6. Limitations and future work

We have presented NFL, a neural field-based approach for synthesizing LiDAR scans from novel viewpoints. NFL combines the benefits of volume rendering with a physically based model of LiDAR acquisition process to faithfully model LiDAR characteristics including beam divergence, secondary returns, and ray dropping. Even though NFL significantly outperforms explicit reconstruct-then-simulate methods as well as other NeRF-style methods, it still has some limitations that we would like to address in future work. Firstly, with NFL, we try to seek a balance between adhering to the physical principles of LiDAR and incorporating semantic features and learning. While our formulation already shows improved performance over baselines, there is still potential for further improvements. For example, while real-world LiDAR sensors perform range detection on the integrated beam radiant, we actually found that using the density-based weights separately for each ray leads to improved performance. Prediction of the second return mask enables us to model secondary returns and to further improve the estimation of the first return. Yet, as indicated by the oracle study, improving the mask prediction could lead to further improvements. Finally, our method is based on a NeRF-style representation and therefore requires per-scene optimization. Generalization across scenes and handling dynamic environments are key challenges that we plan to address in future work.

Acknowledgements.. We sincerely thank Benjamin Naujoks, Steven Butrimas, Tomislav Medić, Yu Han, and Prof. Dr. Andreas Wieser for helpful discussions around LiDAR models. We are grateful for the feedback on figures from Rodrigo Caye Daudt. This appreciation extends to Zvi Greenstein for organisation support.

3.7. Appendix

In this supplementary document, we first present additional information about our dataset, evaluation setting, implementation details in § 3.7.1. We then elaborate on technical details of our methods in § 3.7.3. Additional results of the two return mask segmentation, more quantitative and qualitative results are provided in § 3.7.4.

3.7.1. Datasets and implementation details

Town dataset. To simulate *TownReal* dataset, we approximate a diverged beam profile using 37 subrays and the divergence angle $\gamma_0 = 2$ mrad [12]. We use the subray distribution proposed from [57] (*c.f.* Fig. 3.8). The dataset is shown in Fig. 3.10.

Waymo dataset. We use the following 4 scenes (*c.f.* Fig. 3.11) that are mostly static from *Waymo* [48] dataset

	Scene ID
Scene 1	10017090168044687777_6380_000_6400_000
Scene 2	10096619443888687526_2820_000_2840_000
Scene 3	10061305430875486848_1080_000_1100_000
Scene 4	10275144660749673822_5755_561_5775_561

Waymo NVS setting. We simulate the new trajectory by shifting the sensor by [1.5, 1.5, 0.5] meters (see Fig. 3.11), yielding an overall displacement of ≈ 2.18 meters. This displacement magnitude corresponds to the requirements of various tasks, such as lane changes or adapting the sensor rig from a car to a truck. Moreover, our displacement from the trajectory is similar [59] or even larger [36] than used in prior NVS works. Nevertheless, we run additional experiments by varying the displacements and report results in Tab. 3.6. NFL consistently outperforms baseline methods under different settings, and the improvement is more pronounced under large displacements.

Point cloud registration task. We utilize 49 paired consecutive frames per scene, with a relative displacement of ≈ 1 meter. *TE* is reported in centimeters and *RE* is reported in degrees.

3.7.2. Implementation details

Our method.. Our model is implemented based on *torch-ngp* [52, 34] and can be trained on a single RTX 3090 GPU. During training we minimize 3.6 using the Adam [20] optimiser, with an initial learning rate of 0.005 which linearly decays to 0.0005 towards the end of training. We clip the gradient magnitudes of all parameters to 1.0 to stabilize the optimisation. In the first stage, we sample $N^c = 768$ points and in the second stage $N^f = 64$ points for each ray. The

3. Neural LiDAR Fields for Novel View Synthesis

	i-NGP	DS-NeRF	URF	LiDARsim	Ours
(0.5, 0.5, 0.5)	7.0 / 14.4	7.0 / 16.0	9.0 / 19.6	16.1 / 33.1	5.4 / 13.0
(1.5, 1.5, 1.0)	8.4 / 17.6	7.8 / 18.5	11.0 / 27.5	16.5 / 37.9	5.8 / 14.3
(2.5, 2.5, 1.5)	11.6 / 28.0	9.3 / 22.8	13.9 / 35.5	17.2 / 46.3	6.4 / 18.4

Table 3.6.: Varying the displacement on *Waymo NVS* dataset. Numbers are reported as *MedAE* / *CD* [cm].

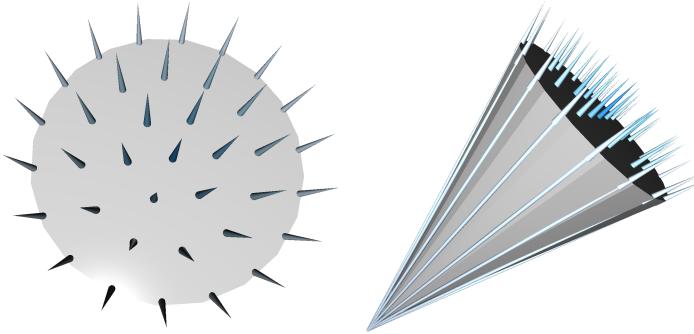


Figure 3.8.: Example diverged beam profile approximated via 37 diverged rays.

window size ϵ for volume rendering is set to 0.8 m, and the buffer value ξ between two returns is set to 2 m. The weights in the loss function, i.e., λ_e , λ_d , and λ_s , are set to 50, 0.15, and 0.15, respectively.

LiDARsim. Because the original implementation is not publicly available, we re-implemented LiDARsim [28] following the paper as close as possible. Specifically, for all points in the training set, we first estimate pointwise normal vectors using all points within a 20 cm radius ball. Then, we apply voxel down-sampling [50] with a voxel size of 4 cm and reconstruct a disk surfel⁷ for each point. Here, the input point represents the disk center and its orientation is defined by the estimated normal vector. At inference time, we perform ray-surfel intersection to determine the intersection points. We empirically observed that LiDARsim’s [28] performance is sensitive to the selected surfel radius. Therefore, we have experimented with both a distance-dependent and fixed surfel radius and found that fixed surfel radius of 6 cm and 12 cm for *Waymo* and *Town* dataset, respectively lead to best range accuracy. To enable second range estimation, we augment LiDARsim with a diverged beam profile approximated using 7 rays. To obtain the second return mask, we consider a LiDAR beam to have two returns if the maximum range difference between all subrays is larger than a threshold⁸. The first return is defined as the closest ray-surfel intersection, while the second return is the nearest one that is at least two meters away. To train the ray drop module, we utilize 40k samples from the Waymo dataset [48], and only apply this module after the ray-surfel intersection to refine the ray drop patterns. Please see Fig. 3.12 for more qualitative results.

Other NeRF methods.. We also use *torch-ngp* [52] codebase to implement other methods,

⁷We use the implementation from Point-Cloud-Utils [56] library.

⁸Sensor-specific parameter, 2 m on *Waymo* dataset.

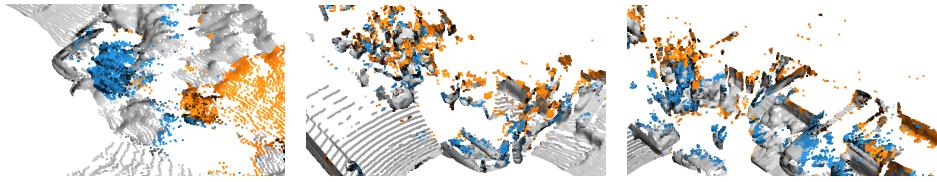


Figure 3.9.: Rendered secondary returns are color-coded in yellow.

using the same network and sampling configurations as used in ours. To estimate the range, we remove the radiance MLP and instead, apply volume rendering of the sampled ζ along the ray. For DS-NeRF [9] and URF [43], we replace their positional encoding with a hash-grid [34] to facilitate a fair comparison with i-NGP [34]. Moreover, we substitute the original L2 loss with the L1 loss, as it results in better performance. Finally, we follow the original paper and augment DS-NeRF [9] and URF [43] with the ray distribution loss and line-of-sight loss, respectively, to regularise the underlying geometry.

3.7.3. Methodology and loss functions

First range estimation. If the maximum weight at the first stage w_p^c is below a predefined threshold $\eta = 0.1$, we assume that the network is uncertain about the reconstruction and the resulting range estimate may be inaccurate. In these cases, we only apply the coarse stage volume rendering and directly estimate the range as: $\hat{\zeta} = \sum_{j=1}^{N_c} w_j^c \cdot \zeta_j$.

Range reconstruction loss. For coarse range, we impose a Gaussian distribution around the ground truth $\hat{\zeta}$ and we anneal the standard deviation δ during training, the annealing procedure is defined as:

$$\delta_k = \delta_{\max} \left(\frac{\delta_{\min}}{\delta_{\max}} \right)^{k/k_{\max}} \quad (3.18)$$

where k denotes the iteration number, k_{\max} is the maximum iteration, and δ_{\max} and δ_{\min} correspond to empirically determined bounds for the standard deviation. The annealing parameters δ_{\min} and δ_{\max} are set to 0.25/0.3 and 1.2/1.6, respectively, for the *Town* and *Waymo* datasets. The maximum iteration k_{\max} is set to 16000/24000 for the *Town* and *Waymo* datasets. The ground truth weight \hat{w}_j is computed as:

$$\hat{w}_j = \int_{\zeta_j}^{\zeta_{j+1}} \frac{1}{\delta \sqrt{2\pi}} \exp \left(-\frac{(x - \hat{\zeta})^2}{2\delta^2} \right) dx. \quad (3.19)$$

3.7.4. Additional results

Runtime analysis. Our *central ray* version takes 4.1 ms per frame to render the single returns on an RTX 3090 GPU, while other NeRF-style methods require 2.4 ms. Only around 10% of rays have second returns, resulting in low computational overhead. While our *diverged beam* incurs additional costs due to querying diverged rays, it can be disabled if needed, without com-

3. Neural LiDAR Fields for Novel View Synthesis

Features			Two return segmentation			Second range		
\bar{f}_{geo}	f_{dir}	f_{range}	Recall \uparrow	Precision \uparrow	IoU \uparrow	Recall@0.5 \uparrow	MAE \downarrow	MedAE \downarrow
✓			78.0	61.6	52.8	60.1	620.1	26.7
✓	✓		79.8	62.9	54.5	61.1	589.1	21.8
✓	✓	✓	82.1	55.6	49.8	67.4	505.1	13.4
threshold depth std.			30.8	24.2	14.8	24.7	1532.2	1461.4

Table 3.7.: Qualitative results of two return segmentation on *Waymo Interp.* dataset.

promising first return performance (*c.f.* Tab. 1). Our re-implementation of LiDARsim achieves 10 Hz runtime, but could be further improved using accelerated ray-tracing, *e.g.* OptiX. Note that all methods already match or even (greatly) exceed the normal LiDAR measurement frequency (≈ 10 Hz).

Ray drop modelling. There clearly is a link between ray drops and beam divergence. However, we found that modeling it through the beam feature yields worse performance, possibly because f_{beam} uses f_{range} , which encodes the statistics of returns and is less meaningful for dropped rays. In future work, beam divergence could instead be incorporated through Intergrated Positional Encoding [4] to model ray drops.

Two return mask prediction. We conduct an ablation study to investigate different design choices for predicting the two return mask and summarize the results in Tab. 3.7. We observe that concatenating the range feature f_{range} with the beam feature f_{beam} improves the segmentation recall and, consequently, the second range estimation. In addition to predicting the two return mask from the beam feature, we experiment with a simple heuristic-based baseline that thresholds the depth standard deviation of sub-rays. Specifically, we considered a LiDAR beam to have two returns if the standard deviation is above 30⁹ cm. However, as shown in Table 3.7, this approach achieves limited success and performs much worse than the learned methods. More qualitative results are presented in Fig. 3.13.

Importance of the second return. Multiple returns are critical for vegetation analysis in remote sensing [24]. NFL is the first work to model the second return by combining beam divergence and *truncated* volume rendering. Unfortunately, second returns do not have semantic annotations in the Waymo dataset, which precluded a quantitative analysis. Nevertheless, qualitatively the rendered second returns are located mostly in vegetation regions, as shown in Fig. 3.9. This correlation suggests that secondary returns could indeed be useful for detecting vegetation.

Semantic segmentation on *Waymo Interp.* dataset. We report additional semantic segmentation results on *Waymo Interp.* dataset in Tab. 3.8. NFL achieves the best performance for vehicle segmentation. Please note that *Waymo Interp.* is of significantly smaller size (10 test frames *vs.*

⁹Empirically determined as it leads to the best Intersection-of-Union score.

Method	Vehicle			Background		
	Recall \uparrow	Precision \uparrow	IoU \uparrow	Recall \uparrow	Precision \uparrow	IoU \uparrow
i-NGP [34] + L2	71.1	97.0	69.4	99.6	96.5	96.2
i-NGP [34]	<u>94.8</u>	89.7	<u>85.6</u>	98.7	<u>99.4</u>	<u>98.1</u>
DS-NeRF [9]	91.4	88.9	82.2	98.7	99.1	97.8
URF [43]	93.8	89.0	84.1	98.6	99.3	97.9
Lidarsim [28]	92.2	74.4	70.2	95.9	99.1	95.1
Ours	95.7	<u>91.2</u>	87.6	<u>98.8</u>	99.5	98.3

Table 3.8.: Semantic segmentation results on *Waymo Interp.* dataset.

Method	TownClean			TownReal			Waymo interp.			Waymo NVS		
	MAE \downarrow	MedAE \downarrow	CD \downarrow	MAE \downarrow	MedAE \downarrow	CD \downarrow	MAE \downarrow	MedAE \downarrow	CD \downarrow	MAE \downarrow	MedAE \downarrow	CD \downarrow
i-NGP [34] + L2	63.6	14.8	37.1	78.2	18.4	44.5	41.4	14.7	24.9	47.3	17.6	29.5
i-NGP [34]	42.2	4.1	17.4	49.8	4.8	19.9	26.4	5.5	11.6	30.4	7.3	<u>15.3</u>
DS-NeRF [9]	<u>41.7</u>	3.9	<u>16.6</u>	<u>48.9</u>	4.4	<u>18.8</u>	<u>28.2</u>	6.3	14.5	30.4	<u>7.2</u>	16.8
URF [43]	43.3	4.2	16.8	52.1	5.1	20.7	28.2	<u>5.4</u>	12.9	43.1	10.0	21.2
LiDARsim [28]	159.6	0.8	23.5	162.8	<u>3.8</u>	27.4	116.3	15.2	27.6	160.2	16.2	34.7
Ours	32.0	<u>2.3</u>	9.0	39.2	3.0	11.5	30.8	5.1	<u>12.1</u>	<u>32.6</u>	5.5	13.2

Table 3.9.: Results of LiDAR novel view synthesis for the first range.

50 frames per scene in other datasets).

Quantitative results. We perform further experiments to evaluate an additional baseline method, denoted as *i-NGP[34] + L2*, which optimizes the range estimation through L2 loss [9, 43]. The comprehensive results of our experimentation are presented in Tab. 3.9, Tab. 3.10, Tab. 3.11, and Tab. 3.12. Our findings reveal that the L2 loss performs inferior to its L1 loss counterpart (*i.e.* i-NGP [34]). However, replacing the standard volume rendering with the proposed formulation for active sensors, still leads to improved performance, as demonstrated in Tab. 3.10.

Qualitative results. We show additional qualitative results in Fig. 3.14, Fig. 3.15, Fig. 3.16, and Fig. 3.17. We sample the middle frame of each dataset and present the first range errors in range-view projection.

3. Neural LiDAR Fields for Novel View Synthesis

Method	TownClean			Waymo Interp.		
	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓
i-NGP [34] + L2	60.8 (-2.8)	12.6 (-2.2)	34.4 (-2.7)	40.8 (-0.6)	13.1 (-1.6)	24.0 (-0.8)
i-NGP [34]	41.0 (-1.2)	4.1 (0.0)	17.6 (0.2)	25.3 (-1.1)	4.5 (-1.0)	10.5 (-1.1)
DS-NeRF [9]	37.4 (-4.2)	3.0 (-0.9)	14.4 (-2.2)	27.4 (-0.8)	5.4 (-1.0)	13.6 (-0.9)
URF [43]	46.4 (3.0)	4.5 (0.3)	18.4 (1.6)	28.3 (0.1)	5.3 (-0.1)	13.1 (0.2)
Ours	32.0 (-2.1)	2.3 (-2.5)	9.0 (-3.9)	30.8 (-2.1)	5.1 (-2.0)	12.1 (-2.3)

Table 3.10.: Ablation study of volume rendering for active sensing.

Method	TownClean			TownReal			Waymo NVS		
	Rec@5 ↑	RE ↓	TE ↓	Rec@5 ↑	RE ↓	TE ↓	Rec@2 ↑	RE ↓	TE ↓
i-NGP [34] + L2	40.6	0.2	6.2	39.6	0.2	6.7	26.5	0.1	3.2
i-NGP [34]	70.3	0.1	4.2	76.0	0.1	4.2	60.2	0.1	1.9
DS-NeRF [9]	58.3	0.2	5.1	56.2	0.2	5.1	42.3	0.1	2.4
URF [43]	61.5	0.2	5.0	59.9	0.1	4.7	32.1	0.1	2.7
LiDARsim [28]	82.8	0.1	3.4	<u>79.2</u>	0.1	3.4	<u>62.8</u>	0.1	<u>1.8</u>
Ours	<u>80.2</u>	<u>0.1</u>	<u>3.7</u>	85.9	<u>0.1</u>	3.4	71.9	0.1	1.7

Table 3.11.: Point cloud registration results on three datasets.

Method	Vehicle			Background		
	Recall ↑	Precision ↑	IoU ↑	Recall ↑	Precision ↑	IoU ↑
i-NGP [34] + L2	68.4	90.2	64.1	99.3	96.3	95.6
i-NGP [34]	<u>93.2</u>	85.9	<u>80.9</u>	98.3	<u>99.2</u>	<u>97.6</u>
DS-NeRF [9]	90.7	<u>87.1</u>	80.2	98.5	98.9	97.4
URF [43]	87.8	81.7	73.7	98.0	98.4	96.5
Lidarsim [28]	90.5	70.5	65.9	94.9	99.0	94.0
Ours	95.9	87.0	83.9	98.3	99.5	97.8

Table 3.12.: Semantic segmentation results on *Waymo NVS* dataset.

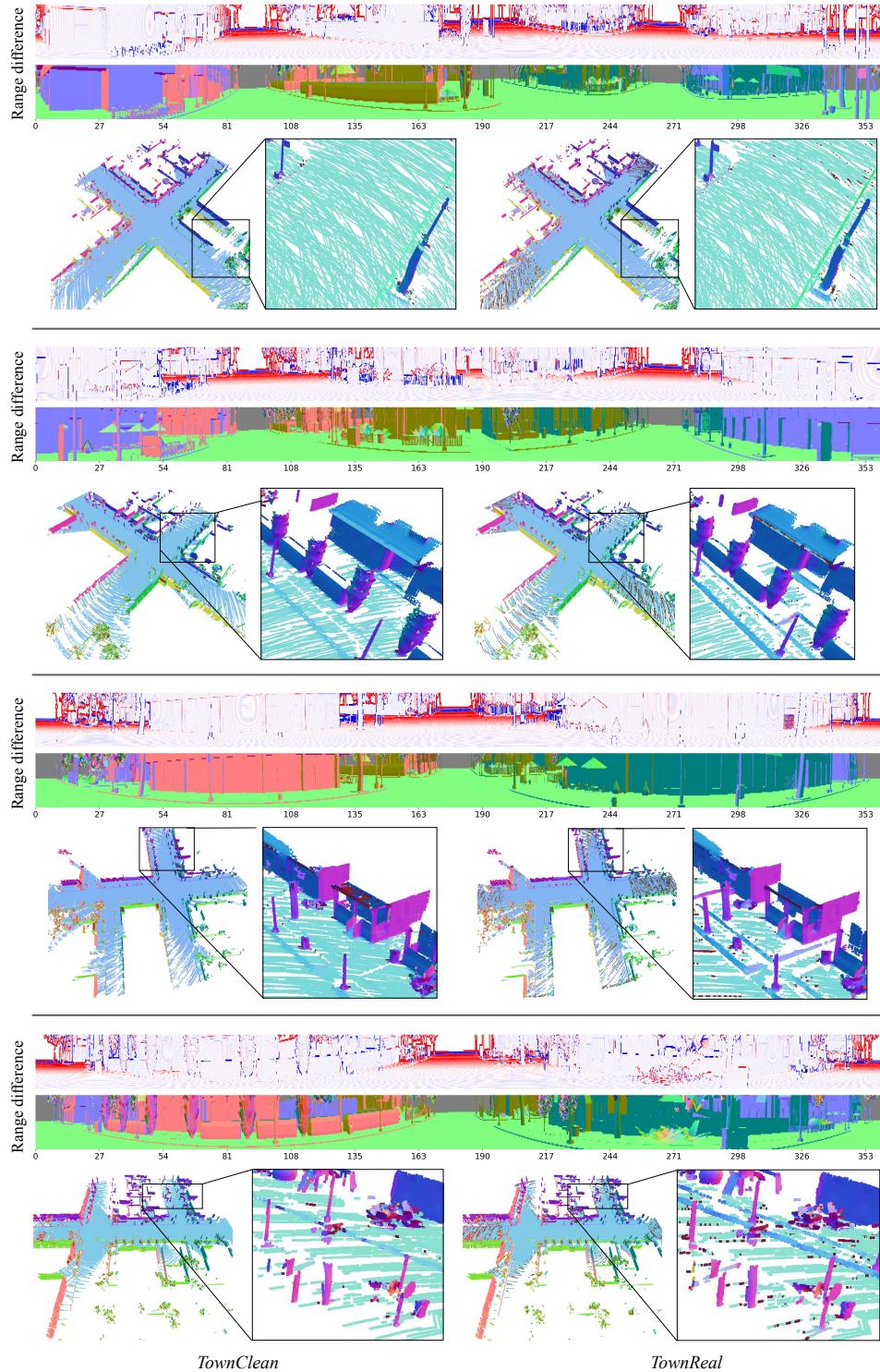


Figure 3.10.: Visualisation of *Town* dataset. Employing a diverged beam profile in range simulation results in an overestimation of range in the high range regime (-16 cm). Such range difference is also reflected on delicate structures, as evidenced by the point cloud view.

3. Neural LiDAR Fields for Novel View Synthesis

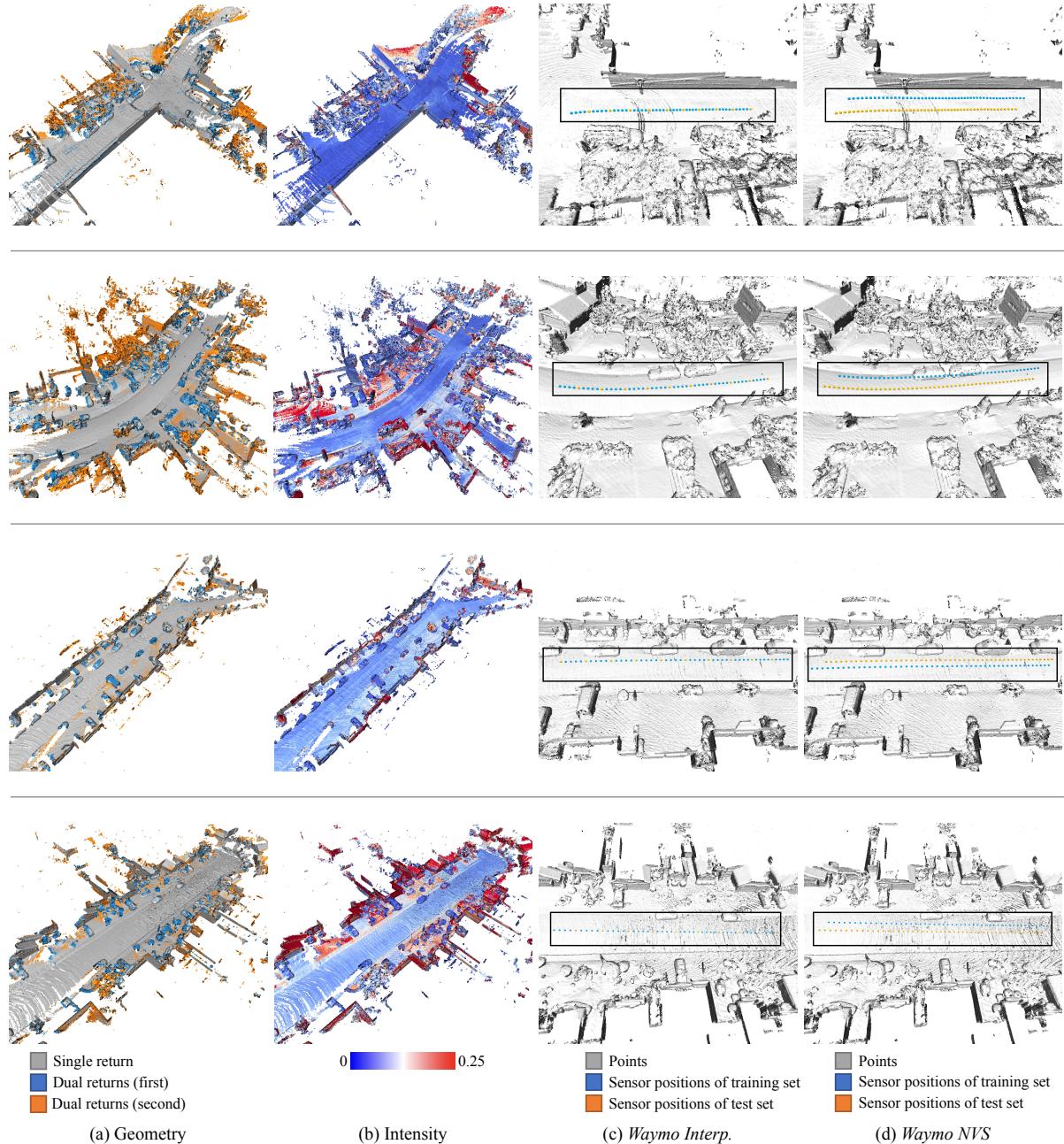


Figure 3.11.: Visualisations of *Waymo* dataset. We accumulate all 50 frames for each scene and show their geometry, intensity profile, and sensor positions of training and test sets on *Waymo Interp.* and *Waymo NVS* datasets.

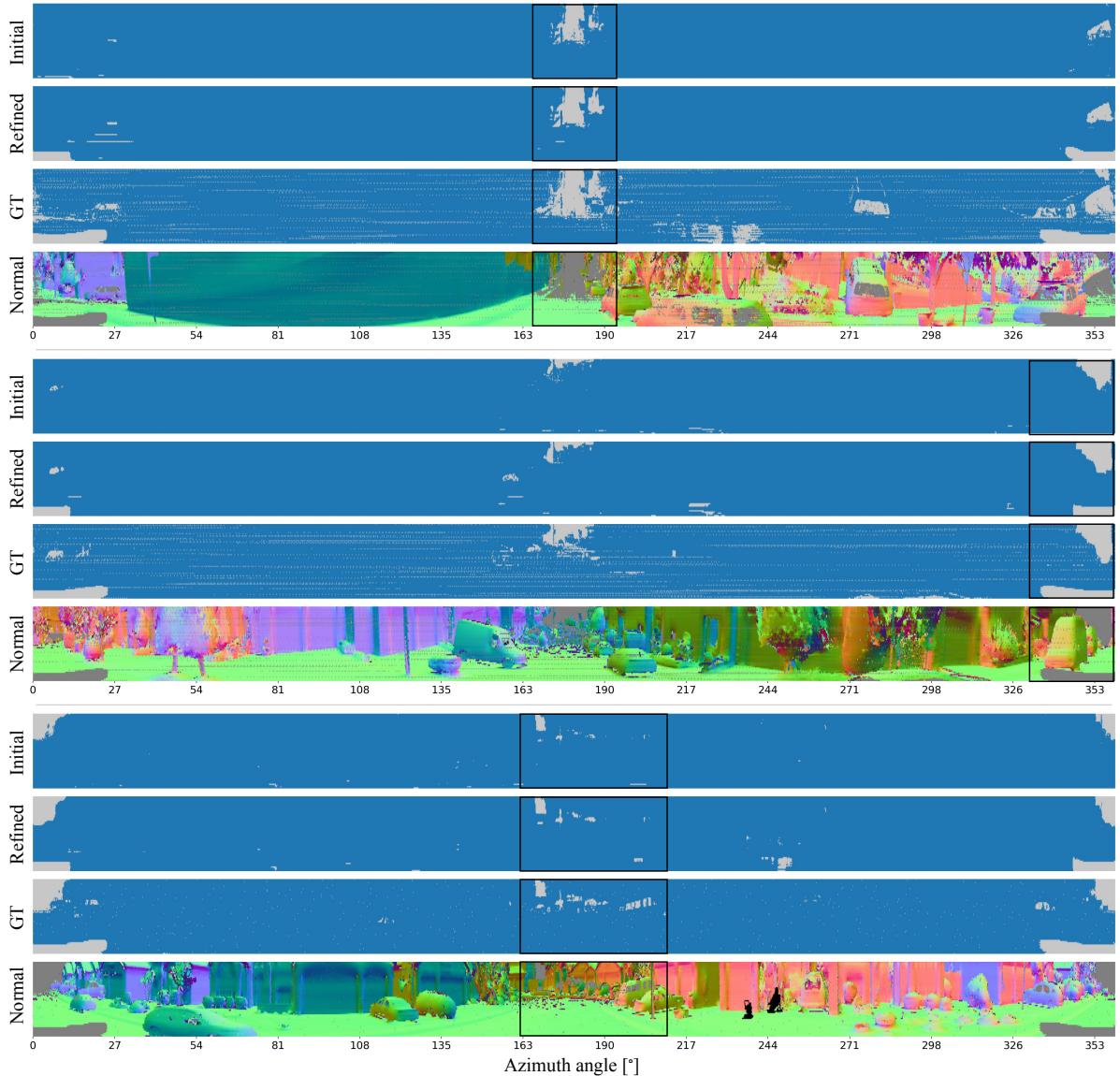


Figure 3.12.: Ray drop segmentation on *Waymo Interp.* dataset using LiDARsim [28]. We show both the initial ray drop mask from ray-surfel query and the refined masks using learned ray-drop model.

3. Neural LiDAR Fields for Novel View Synthesis

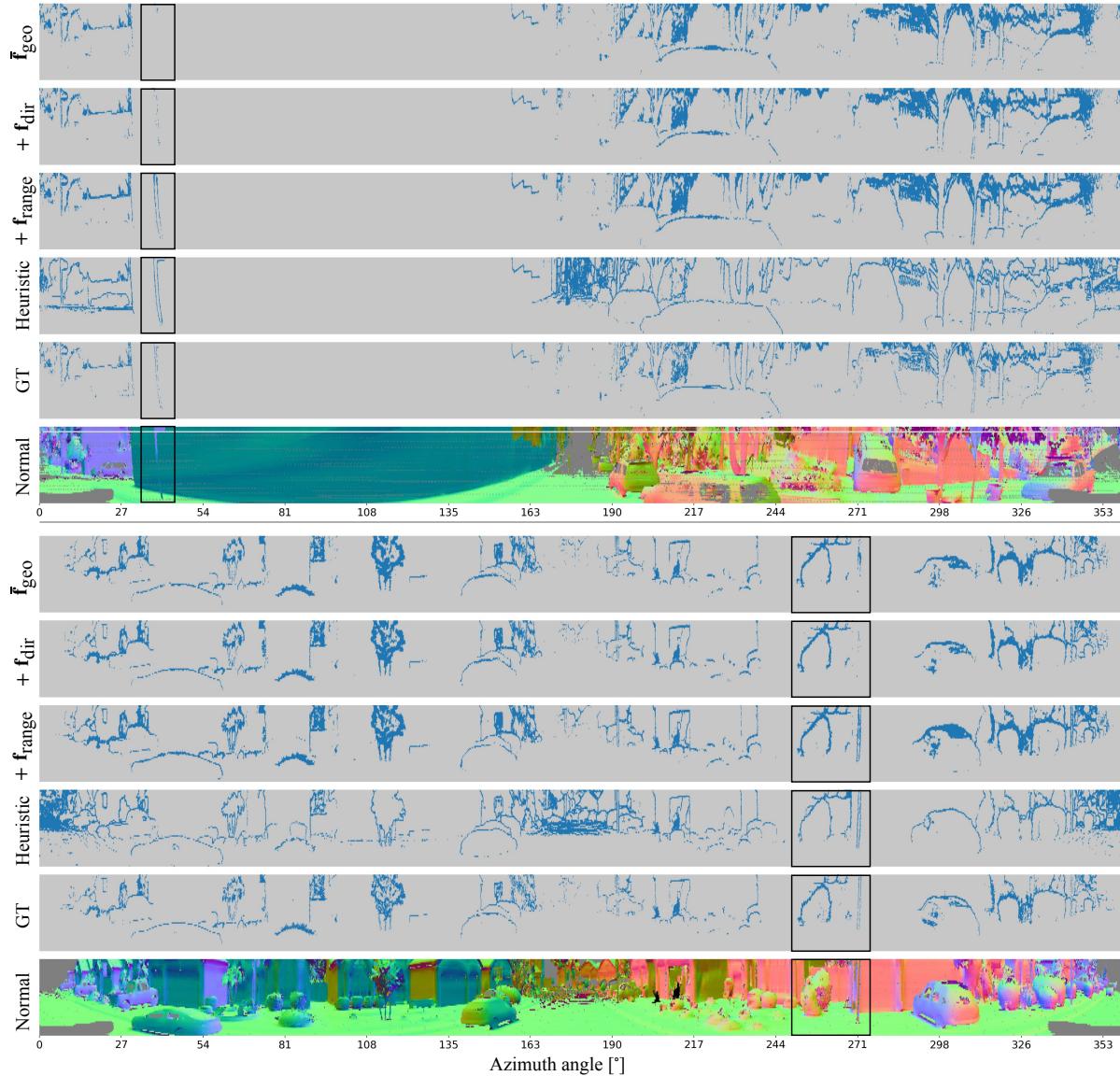
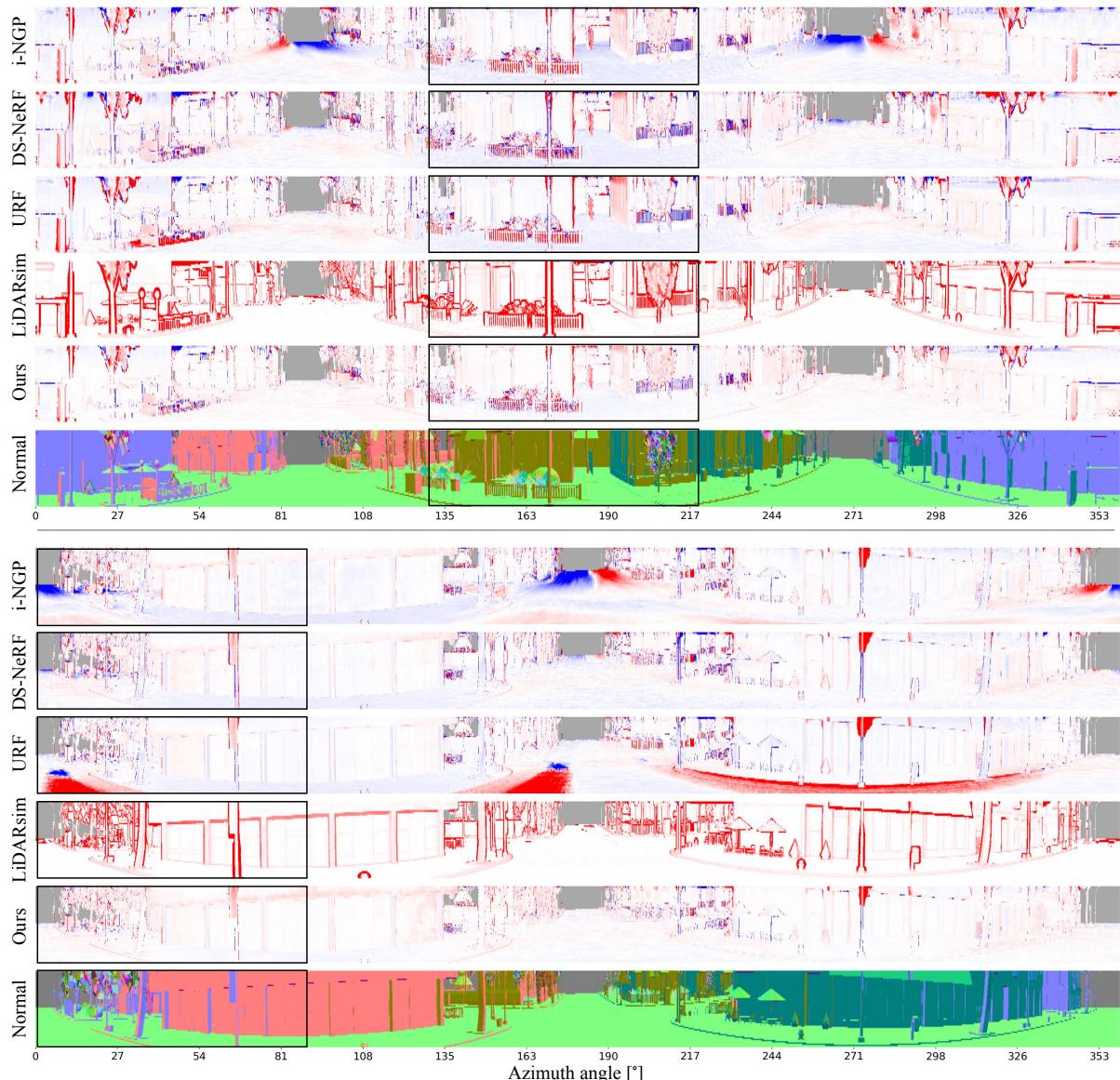


Figure 3.13.: Qualitative results of two return mask segmentation.

Figure 3.14.: Qualitative results of first range estimation on *TownClean* dataset.

3. Neural LiDAR Fields for Novel View Synthesis

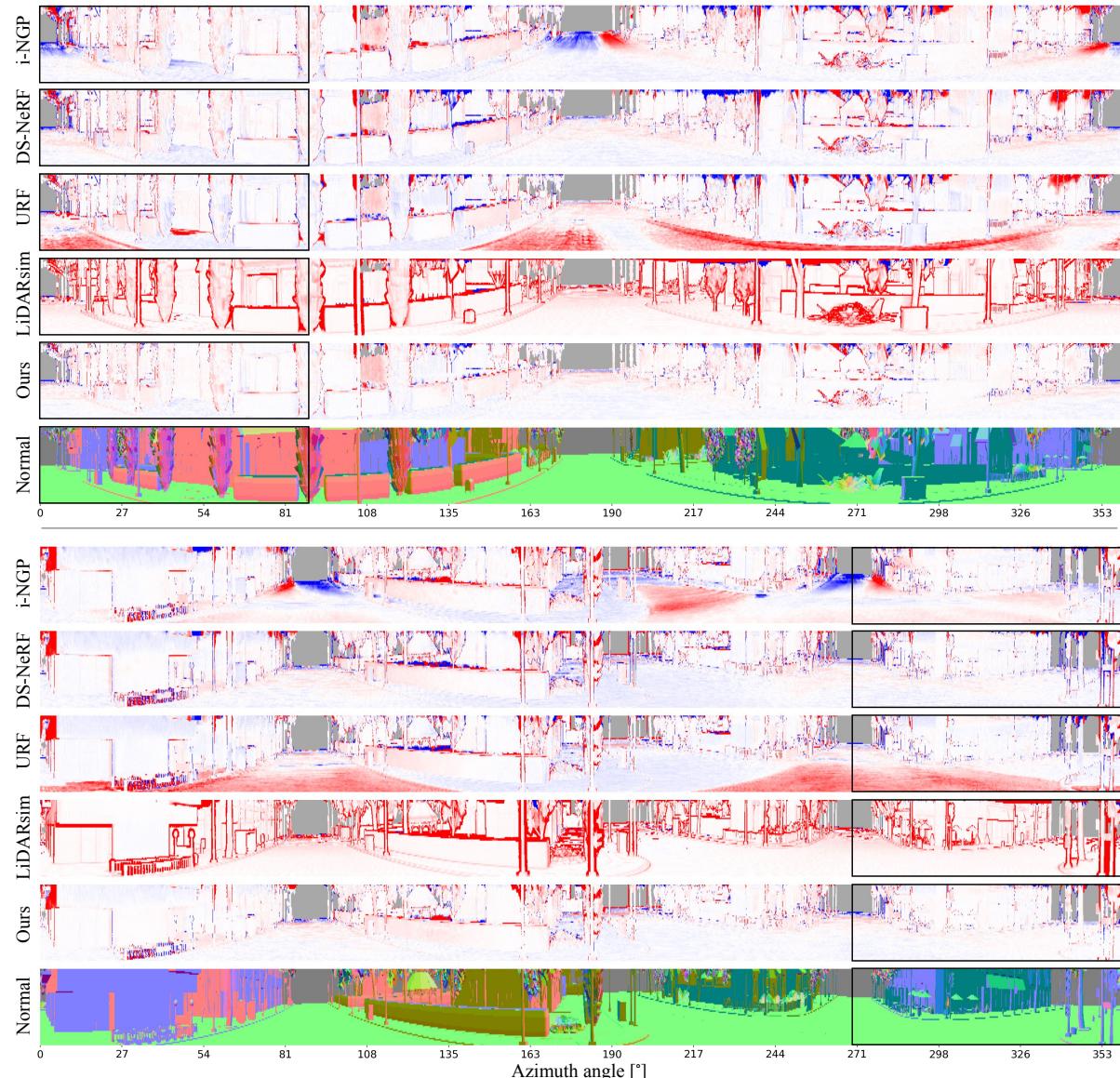
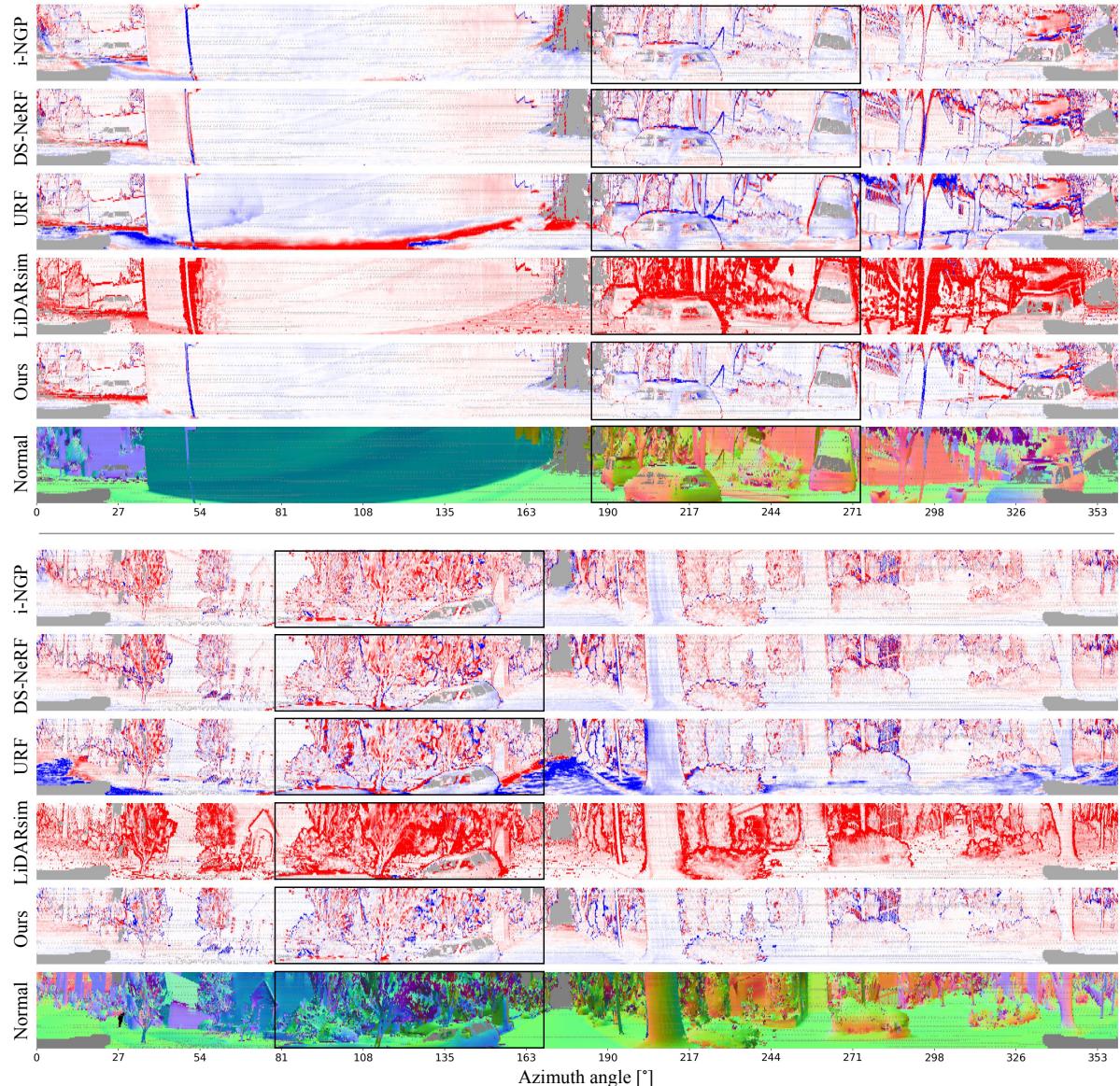


Figure 3.15.: Qualitative results of first range estimation on *TownReal* dataset.

Figure 3.16.: Qualitative results of first range estimation on *Waymo NVS* dataset.

3. Neural LiDAR Fields for Novel View Synthesis

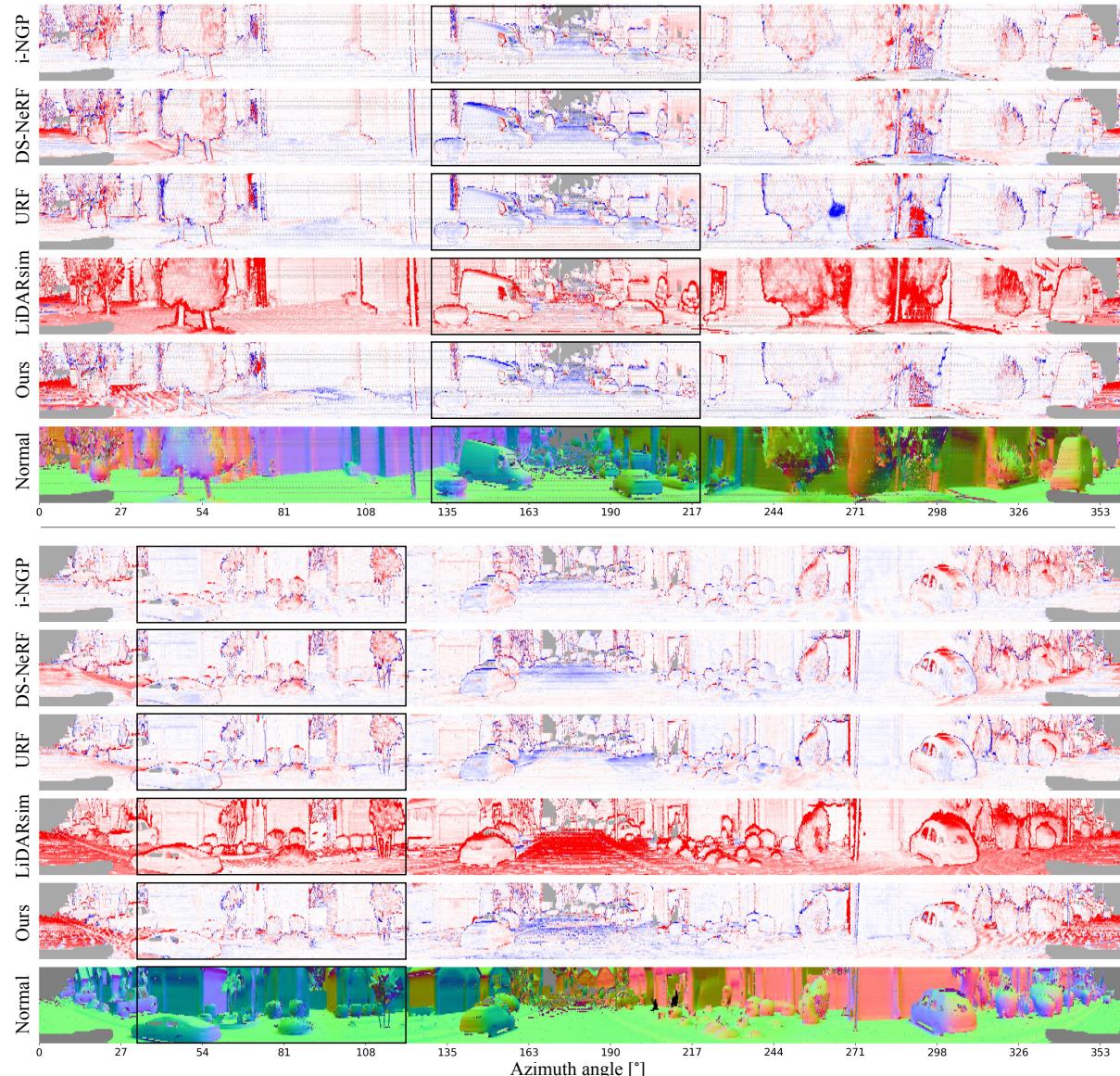


Figure 3.17.: Qualitative results of first range estimation on *Waymo Interp.* dataset.

4 | Conclusions

placeholder

4.1. Lessons Learned

placeholder

4.2. Limitations

placeholder

4.3. Outlook

placeholder

A | Bibliography

- [1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O’Toole. TöRF: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.
- [5] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [6] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [7] Tomas Carlsson, Ove Steinvall, and Dietmar Letalick. Signature simulation and signal analysis for 3-d laser radar. Technical report, Swedish Defence Research Agency, 2001.
- [8] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. GARF: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv preprint arXiv:2204.05735*, 2022.
- [9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021.
- [10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*. PMLR, 2017.
- [11] Jin Fang, Dingfu Zhou, Feilong Yan, Tongtong Zhao, Feihu Zhang, Yu Ma, Liang Wang, and Ruigang Yang. Augmented LiDAR simulator for autonomous driving. *IEEE RA-L*, 5(2):1931–1938, 2020.

A. Bibliography

- [12] Craig Glennie. Calibration and kinematic analysis of the Velodyne HDL-64E S2 lidar sensor. *Photogrammetric Engineering & Remote Sensing*, 78(4):339–347, 2012.
- [13] Benoit Guillard, Sai Vemprala, Jayesh K Gupta, Ondrej Miksik, Vibhav Vineet, Pascal Fua, and Ashish Kapoor. Learning to simulate realistic LiDARs. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [14] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. LiDAR snowfall simulation for robust 3d object detection. *arXiv preprint arXiv:2203.15118*, 2022.
- [15] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real LiDAR point clouds for 3d object detection in adverse weather. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [16] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3d scene analysis by point cloud accumulation. In *European Conference on Computer Vision (ECCV)*, 2022.
- [17] Shengyu Huang, Zan Gojcic, Mikhail Usyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [18] Kasiopy. Town with suburb. <https://www.turbosquid.com/3d-models/town-suburb-3d-max/1085661>. last accessed 2023.
- [19] Velat Kilic, Deepti Hegde, Vishwanath Sindagi, A Brinton Cooper, Mark A Foster, and Vishal M Patel. Lidar light scattering augmentation (LISA): Physics-based simulation of adverse weather conditions for 3d object detection. *arXiv preprint arXiv:2107.07004*, 2021.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.
- [21] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-NeRF: Neural radiance fields from a moving event camera. *arXiv preprint arXiv:2208.11300*, 2022.
- [22] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS)*, 2004.
- [23] Akhil Kurup and Jeremy Bos. DSOR: A scalable statistical filter for removing falling snow from LiDAR point clouds in severe winter weather. *arXiv preprint arXiv:2109.07078*, 2021.
- [24] Kevin Lim, Paul Treitz, Michael Wulder, Benoît St-Onge, and Martin Flood. Lidar remote sensing of forest structure. *Progress in physical geography*, 27(1):88–106, 2003.
- [25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

- [26] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [27] Andrew Luo, Yilun Du, Michael J Tarr, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *arXiv preprint arXiv:2204.00628*, 2022.
- [28] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. LiDARsim: Realistic LiDAR simulation by leveraging the real world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [29] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [30] Nelson Max and Min Chen. Local and global illumination in the volume rendering integral. Technical report, Lawrence Livermore National Lab., Livermore, CA, 2005.
- [31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [32] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NerF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.
- [34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.
- [35] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [36] Julian Ost, Issam Laradji, Alejandro Newell, Yuval Bahat, and Felix Heide. Neural point light fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [37] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

A. Bibliography

- [39] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Computer Graphics and Interactive Techniques*, 2000.
- [40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [41] Mohamad Qadri, Michael Kaess, and Ioannis Gkioulekas. Neural implicit surface reconstruction using imaging sonar. *arXiv preprint arXiv:2209.08221*, 2022.
- [42] Ralph H Rasshofer, Martin Spies, and Hans Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9:49–60, 2011.
- [43] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. *arXiv preprint arXiv:2111.14643*, 2021.
- [44] Barbara Roessler, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. *arXiv preprint arXiv:2112.03288*, 2021.
- [45] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Event-NeRF: Neural radiance fields from a single colour event camera. *arXiv preprint arXiv:2206.11896*, 2022.
- [46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [47] Yi-Chien Shih, Wei-Hsiang Liao, Wen-Chieh Lin, Sai-Keung Wong, and Chieh-Chih Wang. Reconstruction and synthesis of lidar point clouds of spray. *IEEE RA-L*, 7(2):3765–3772, 2022.
- [48] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [49] Andrea Tagliasacchi and Ben Mildenhall. Volume rendering digest for nerf. *arXiv preprint arXiv:2209.02417*, 2022.
- [50] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *MLSys*, volume 4, pages 302–315, 2022.
- [51] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision (ECCV)*, 2020.
- [52] Jiaxiang Tang. Torch-ngp: a PyTorch implementation of instant-ngp, 2022. <https://github.com/ashawkey/torch-ngp>.

- [53] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision (ECCV)*, 2014.
- [54] Wolfgang Wagner, Andreas Ullrich, Vesna Ducic, Thomas Melzer, and Nick Studnicka. Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(2):100–112, 2006.
- [55] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
- [56] Francis Williams. Point cloud utils, 2022. <https://www.github.com/fwilliams/point-cloud-utils>.
- [57] Lukas Winiwarter, Alberto Manuel Esmorís Pena, Hannah Weiser, Katharina Anders, Jorge Martínez Sánchez, Mark Searle, and Bernhard Höfle. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. *Remote Sensing of Environment*, 269:112772, 2022.
- [58] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.
- [59] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [60] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [61] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.
- [62] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

B | List of Publications

1. **S. Huang**, M. Usvyatsov, K. Schindler. "Indoor Scene Recognition in 3D." IROS, 2020.
2. **S. Huang***, Z. Gojcic*, M. Usvyatsov, A. Wieser, K. Schindler. "PREDATOR: Registration of 3D Point Clouds with Low Overlap." CVPR, 2021. (Oral)
3. **S. Huang**, Z. Gojcic, J. Huang, A. Wieser, K. Schindler "Dynamic 3D Scene Analysis by Point Cloud Accumulation." ECCV, 2022.
4. **S. Huang**, Z. Gojcic, Z. Wang, F. William, Y. Kasten, S. Fidler, K. Schindler, O. Litany "Neural LiDAR Fields for Novel View Synthesis." ICCV, 2023.
5. H. Wu, X. Zuo, S. Leutenegger, Or Litany, K. Schindler, **S. Huang** "Dynamic LiDAR Re-simulation using Compositional Neural Fields." CVPR, 2024. (Highlight)
6. L. Zhu, **S. Huang**, K. Schindler, I. Armeni. "Living Scenes: Multi-object Relocalization and Reconstruction in Changing 3D Environments." CVPR, 2024. (Highlight)
7. B. Ke, A. Obukhov, **S. Huang**, N. Metzger, R. Daudt, K. Schindler. "Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation." CVPR, 2024. (Oral, Best Paper Award Candidate)
8. Y. Jia, L. Hoyer, **S. Huang**, T. Wang, L. Gool, K. Schindler, A. Obukhov. "DGInStyle: DomainGeneralizable Semantic Segmentation with Image Diffusion Models and Stylized Semantic Control." ECCV, 2024.
9. Z. Wang, T. Shen, J. Gao, **S. Huang**, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, S. Fidler "Neural Fields meet Explicit Geometric Representations for Inverse Rendering of Urban Scenes." CVPR, 2023.
10. L. Zhu, Y. Jia, **S. Huang**, N. Meyer, A. Wieser, K. Schindler, J. Aaron "DeFlow: Self-supervised 3D Motion Estimation of Debris Flow." CVPR Workshop, 2023. (Best Paper Award)
11. C. Stucker, B. Ke, Y. Yue, **S. Huang**, I. Armeni, K. Schindler "ImpliCity: City Modeling from Satellite Images with Deep Implicit Occupancy Fields." ISPRS Congress, 2022. (Best Young Author Award)
12. T. Sun, Y. Hao, **S. Huang**, S. Savarese, K. Schindler, M. Pollefeys, I. Armeni "Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change." arxiv, 2023

C | Curriculum Vitae

Personal data

Name **Shengyu Huang**

Date of birth **17.11.1995**

Nationality **China**

Education

Oct. 2020 - Nov. 2024 **ETH Zurich, Switzerland**
PhD Student

Sep. 2018 - Aug. 2020 **ETH Zurich, Switzerland**
M.Sc. in Geomatik

Sep. 2014 - Aug. 2018 **Tongji University, China**
BSc. in Surveying and Mapping Engineering

Professional Experience

Oc. 2020 - Nov. 2024 **ETH Zurich, Switzerland**
Research and Teaching Assistant

Jul 2023 - Dec 2023 **Google, Munich, Germany**
Student Researcher

Apr 2022 - Dec 2022 **NVIDIA, Zurich, Switzerland**
Research Scientist Intern