

# Self-Supervised Depth Correction of Lidar Measurements from Map Consistency Loss

Ruslan Agishev, Tomáš Petříček, Karel Zimmermann

**Abstract**—Depth perception is considered an invaluable source of information in the context of 3D mapping and various robotics applications. However, point cloud maps acquired using consumer-level *light detection and ranging* sensors (lidars) still suffer from bias related to local surface properties such as measuring beam-to-surface incidence angle. This fact has recently motivated researchers to exploit traditional filters, as well as the deep learning paradigm, in order to suppress the aforementioned depth sensors error while preserving geometric and map consistency details. Despite the effort, depth correction of lidar measurements is still an open challenge mainly due to the lack of clean 3D data that could be used as ground truth. In this paper, we introduce two novel point cloud map consistency losses, which facilitate self-supervised learning on real data of lidar depth correction models. Specifically, the models exploit multiple point cloud measurements of the same scene from different view-points in order to learn to reduce the bias based on the constructed map consistency signal. Complementary to the removal of the bias from the measurements, we demonstrate that the depth correction models help to reduce localization drift. Additionally, we release a data set that contains point cloud scans captured in an indoor corridor environment with precise localization and ground truth mapping information<sup>1</sup>.

**Index Terms**—Range Sensing, 3D Mapping, LIDAR, Data Sets for SLAM

## I. INTRODUCTION

Lidars has been deployed in a myriad of applications, including but not limited to *simultaneous localization and mapping* (SLAM) for mobile robots and autonomous cars, and remote sensing ([13], [19], [17]). However, point cloud maps acquired using consumer-level *light detection and ranging* sensors (lidars) still suffer from various biases induced by immediate capturing conditions. It is well-known that different environmental conditions such as illumination, temperature or humidity [7] and surface reflectance [10] affect lidar performance. A lidar measurements accuracy could also be influenced by diverse material types and scanned distance (see and

Manuscript received: February, 27, 2023; Revised May, 13, 2023; Accepted June, 12, 2023.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

This work was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22\_008/0004590), by Czech Science Foundation under Project 20-29531S, by OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16\_019/0000765 "Research Center for Informatics", and by the European Union's Horizon Europe Framework Programme under the euRobin project (ID: 101070596). R. Agishev and T. Petříček were supported by Grant Agency of the CTU Prague under Project SGS22/111/OHK3/2T/13. (Corresponding author: R. Agishev.)

The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 166 36 Prague, Czech Republic (e-mail: agishrus@fel.cvut.cz; tpetricek@gmail.com; zimmerk@fel.cvut.cz)

Digital Object Identifier (DOI): see top of this page.

<sup>1</sup>[http://ptak.felk.cvut.cz/vras/data/fec\\_corridor/](http://ptak.felk.cvut.cz/vras/data/fec_corridor/)



Fig. 1. Mobile robot with lidar mapping a corridor-like environment. Points color denote lidar beams incidence angle (darker color - higher angle). Measurements with higher incidence angle have higher bias [9]. The snapshot is taken from the FEE Corridor data set (Sec. IV-B). The data set is recorded for learning the lidar measurements correction models as well as to evaluate SLAM and reconstruction algorithms.

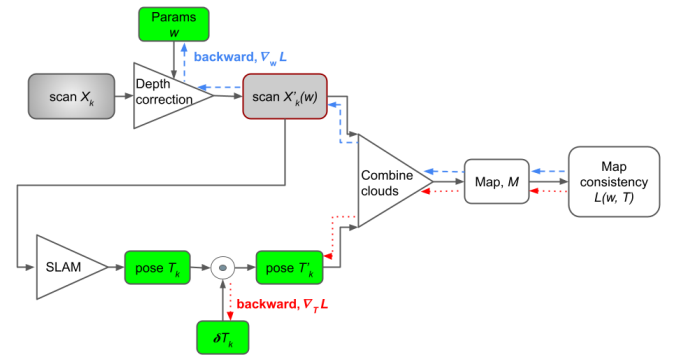


Fig. 2. Self-supervised pipeline to learn lidar measurements correction model and optimize sensor poses. Point cloud map is constructed from corrected lidar measurements and sensor poses. We compute the map consistency loss on the resultant map and propagate the error back to model parameters as well as to corresponding poses correction terms. The depth correction model removes the bias from raw lidar scans.

example in Figure 5). In this work we confirm that a substantial bias typically emerges when measuring surfaces with a high incidence angle (Figure 1). The resulting measurement bias then negatively influences both the localization and mapping accuracy; therefore, some kind of bias-compensation module is highly desirable.

In many applications, such as the outdoor deployment of a mobile robotic platform, the accurate capturing conditions are a priori unknown and dynamically evolving. In such situations, the bias-compensating module has to be regularly (or even continuously) adapted on immediate capturing conditions. Since reliable measuring of exact capturing conditions would

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

require various exotic sensors onboard, we instead focus on a self-contained solution independent of other sensor modalities that can naturally adapt the bias-compensating module just from lidar measurements. Since there are no ground truth depths that can be used for supervised adaptation, an alternative source of self-supervision is required.

In the task of lidar-based SLAM, the natural expectations are that the resulting map will look like a map of a typical environment. In order to perform the adaptation, we address the two underlying sub-problems: (i) What is a suitable self-supervised loss that measures the inconsistency between reality and expectations? (ii) What is the structure of the depth-correcting module that allows for end-to-end learning from this self-supervised loss?

Towards this end, we propose the fully-differentiable depth-correcting module, which leverages a state-of-the-art first-principle model of lidar-surface interaction. We also introduce a so-called *map consistency loss* - the measure that encodes our prior assumptions about the appearance of the typical environment. There is a trade-off between the strength of the prior assumptions and the universality of the solution: the more substantial the assumptions, the better the self-supervision, but the lower the variability of environments the robot can operate in. Since we target on a generally applicable solution, we introduce weaker assumptions that are satisfied in an almost arbitrary environment. We experimentally demonstrate that even these weak assumptions are sufficient for the successful self-supervised learning of the bias/depth-compensating module since the resulting approach increases the localization and mapping accuracy of a typical SLAM method.

The **key contributions** of the work are following.

- Novel point cloud map consistency losses are designed, which could be used in a self-supervised manner for depth measurements correction, as well as sensors calibration tasks.
- A self-supervised training method is introduced aimed to learn lidar measurements correction models.
- It is demonstrated that utilization of the depth correction models not only allows to suppress the bias from lidar measurements, but also to create more accurate maps, and reduce localization drift in SLAM scenarios.
- Novel data sequences are released which contain accurately localized point cloud data with a ground truth map.

The paper is organized as follows. Section II presents related work. The description of depth correction models and map consistency loss is given in Section III. Experiments and evaluation studies are described in Section IV. Finally, conclusions are drawn in Section V. To facilitate reproducibility of the paper, we provide the source code<sup>2</sup>.

## II. RELATED WORK

A laser beam width is typically modeled as a Gaussian function of the distance along the beam [3]. Using non-zero-mean Gaussian beam model, Laconte *et al.* [9] were able to measure the bias of several commercial lidar sensors as a

function of depth and incidence angle. The authors additionally present an experimental methodology on estimation of the bias model parameters for a lidar sensor. We provide an extension of their work, by introducing a learning method to estimate the bias model parameters using raw sensory measurements. Sterzentsenko *et al.* [16] presented self-supervised depth denoising method for RGB-D sensors. Similar to our approach, the authors support the idea of utilizing spatial consistency and surface smoothness as optimization objectives. In our work, the lidar depth correction model is used that was introduced in [8]. It estimates the bias as a polynomial function of laser beam incidence angle, Section III-A. Additionally, we propose a new model that includes dependence of the bias on measured distance by a laser beam (Section III-A). This idea is motivated by the depth measurements correction results presented in [9]. The authors of [8] utilize the model for lidars and cameras calibration scenario. However, the application of the model in localization and mapping scenarios was not studied in their paper.

As we pose the task of global point cloud map consistency estimation as optimization problem, a differentiable loss function is required as a minimization objective. The Chamfer loss introduced in [6] is one of the most famous differentiable function for point sets. It is usually utilized to evaluate a distance between two point clouds, thus is typically suitable for supervised tasks. It was shown in [1] that the introduced quadric loss in combination with the Chamfer loss improves 3D reconstruction accuracy by preserving sharp edges of scene objects. The loss is designed to reduce quadric error between the reconstructed points and the input surface. However, as it was mentioned by the authors of [1], the quadric loss can not preserve the input point distribution, since the quadric loss is zero anywhere on the plane. On the contrary, in order to reduce the lidar measurements bias caused by high incidence angle, we target to correct mainly the points which belong to planar surfaces. Therefore a designed global map consistency loss should be able to encourage points from different scans that represent such objects as walls, floors, and roads to belong to the same plane. Moreover, as shown in the next Section III, we do not require ground-truth data to evaluate the point cloud consistency, and the introduced losses are computed directly on the global map independently on the number of lidar scans and view-points forming the input point cloud.

## III. METHODOLOGY

The method is outlined in Figure 2. Briefly, a depth correction model is applied to individual laser scans (point clouds). The corrected point clouds are transformed into a global reference frame (that of the first measurement) to build a global point cloud map. Then, a consistency loss is computed for this map and its gradient is propagated back in the computation graph to update model learnable parameters as well as to correct sensor poses. We discuss further data representation, models choice, optimization algorithm as well as map consistency functions used as optimization criteria.

### A. Depth Correction Model

<sup>2</sup>[https://github.com/ctu-vras/depth\\_correction](https://github.com/ctu-vras/depth_correction)

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

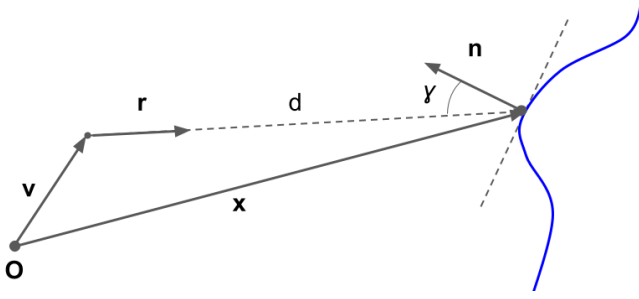


Fig. 3. Lidar scan points representation. A ray represented by vector  $\mathbf{r}$  falls at an oriented surface (drawn as a blue curve);  $\mathbf{v}$  is the vector pointing to the ray origin (sensor location),  $\mathbf{x}$  is a vector defining the measured point location. The surface normal  $\mathbf{n}$  at the beam heating point forms the lidar ray incidence angle  $\gamma$  with the vector  $-\mathbf{r}$ .

Lidar systems are based on laser ranging, which measures the distance between a sensor and a target based on half the elapsed time between the emission of a pulse and the detection of a reflected return [2]. The systems are classified as either discrete return (record single or multiple returns from a given laser pulse) or full waveform recording (digitize the entire reflected energy) [19]. In discrete return model, as the laser signal is reflected back to the sensor, large peaks, (i.e., bright returns), are interpreted as objects in the path of the beam. This makes them favourable for 3D mapping applications, where the sensed environment is represented by clouds of points characterising intercepted features. For the real lasers, the light beam width is not constant along its propagation axis [18]. That is why the intensity distribution is not uniform along a beam direction.

When the laser beam hits an oriented surface, a great amount of intensity returns sooner to the sensor than if the surface was not oriented. This means that the returned waveform will have a different shape depending on the incidence angle, reflectance, and distance inducing a bias in the measurement [9]. Since we focus on a self-contained system, which will be independent of other sensors that would allow to predict surface properties directly, the proposed depth-correcting module infer the immediate bias based on the incidence angle and measured depth.

Lidar measurement (a scan) is represented as a set of points  $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^{3 \times 1}$ . Each point  $\mathbf{x}$  is expressed as

$$\mathbf{x} = \mathbf{v} + d\mathbf{r}, \quad (1)$$

in terms of origin  $\mathbf{v} \in \mathbb{R}^{3 \times 1}$  of the measuring ray, its direction  $\mathbf{r} \in \mathbb{R}^{3 \times 1}$ ,  $\|\mathbf{r}\| = 1$ , and depth  $d > 0$ , Figure 3. For normal vector  $\mathbf{n} \in \mathbb{R}^{3 \times 1}$ ,  $\|\mathbf{n}\| = 1$ , which is perpendicular to the surface at point  $\mathbf{x}$  and pointing outward from the object, incidence angle is given by

$$\gamma = \arccos(-\mathbf{n}^\top \mathbf{r}). \quad (2)$$

Following [9], we estimate depth bias  $\epsilon$  from measured depth  $d$  and incidence angle  $\gamma$ , resulting in corrected depth

$$\hat{d} = d - \epsilon(d, \gamma), \quad (3)$$

and corrected point  $\hat{\mathbf{x}} = \mathbf{v} + \hat{d}\mathbf{r}$ . We evaluate the following depth correction models: the polynomial model introduced in [8],

$$\epsilon_p(\gamma; w_1, w_2) = w_1\gamma^2 + w_2\gamma^4, \quad (4)$$

and the polynomial model scaled by depth,

$$\epsilon_{sp}(d, \gamma; w_1, w_2) = d(w_1\gamma^2 + w_2\gamma^4), \quad (5)$$

where  $w_1, w_2 \in \mathbb{R}$  are the learnable parameters of the models. In the following, we will denote  $w$  the learnable parameters of the model and  $\hat{\mathcal{X}}(w)$  the set of points  $\mathcal{X}$  corrected using a model with parameters  $w$ .

### B. Map Consistency Loss Functions

Let us denote  $\mathcal{N}_{\mathcal{X}}(\mathbf{x}, r)$  the set of points  $\mathbf{x}' \in \mathcal{X}$  within distance  $r$  from a given point  $\mathbf{x}$ . For a set of points  $\mathcal{X}$ , let us denote  $\bar{\mathbf{x}}$  the sample mean,  $\bar{\mathbf{x}} = 1/|\mathcal{X}| \sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x}$ , and  $Q$  the sample covariance matrix,

$$Q = \frac{1}{|\mathcal{X}| - 1} \sum_{\mathbf{x} \in \mathcal{X}} (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^\top. \quad (6)$$

Note that  $Q$  is a  $3 \times 3$  real symmetric matrix. We denote  $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \in \mathbb{R}$  its eigenvalues and  $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in \mathbb{R}^{3 \times 1}$ ,  $\|\mathbf{u}_i\| = 1$ ,  $\mathbf{u}_i^\top \mathbf{u}_{j \neq i} = 0$ , their corresponding eigenvectors, such that  $A\mathbf{u}_i = \lambda_i \mathbf{u}_i$  for  $i = 1, \dots, 3$ .

In *principal component analysis* (PCA), the eigenvectors of a sample covariance matrix are also referred to as principal components. They form an orthogonal basis in which the individual dimensions of the data are linearly uncorrelated and which allows to form lower-dimensional projections of the data that preserve as much variance as possible in the given number of dimensions. Eigenvalue  $\lambda_i$  corresponds to the variance of the data projected onto the corresponding eigenvector  $\mathbf{u}_i$ .

We propose two distinct map consistency loss functions, generically denoted  $\ell$ , which are defined as functions of the sample covariance matrix  $Q$  from (6):

$$\lambda_1, \text{ its minimum eigenvalue, and} \quad (7)$$

$$\text{tr } Q, \text{ the sum of elements on its main diagonal.} \quad (8)$$

The former corresponds to the variance along the direction in which the observed variance is minimum, i.e., along eigenvector  $\mathbf{u}_1$ . Eigenvector  $\mathbf{u}_1$  oriented toward the sensor origin  $\mathbf{v}$  is also used as an estimate of surface normal vector,  $\hat{\mathbf{n}} = -\text{sgn}(\mathbf{u}_1^\top \mathbf{r})\mathbf{u}_1$ . Since trace of a matrix corresponds to the sum of its eigenvalues, the latter loss function corresponds to the sum of variances along the three orthogonal directions given by eigenvectors  $\mathbf{u}_i$ ,  $\text{tr } Q = \sum_{i=1}^3 \lambda_i$ .

One can see similarities with optimization criteria commonly used in iterative closest point (ICP) algorithms. Point-to-plane ICP [5] is similar to (7) in the sense that it only penalizes mis-alignment in the direction perpendicular to the surface while point-to-point ICP [4] penalizes mis-alignment in any direction, similarly to (8). Unlike these and other ICP criteria [15], both proposed losses are formulated in terms of the resulting map and are thus independent of the order or the number of contributing input point clouds. Moreover, the

sampling points for which we gather the neighboring points, construct the covariance matrices, and compute the loss can also be chosen independently of input point clouds.

### C. Self-supervised Learning of Depth Correction

In the following, a point  $\mathbf{x} \in \mathbb{R}^{3 \times 1}$  transformed by rigid transformation  $T$  will be written simply as  $T\mathbf{x}$  while a set of points  $\mathcal{X} \subset \mathbb{R}^{3 \times 1}$  which are transformed by the same transformation will be written as  $T\mathcal{X}$ . Rigid transformation parameterized by vector  $\mathbf{p} \in \mathbb{R}^{6 \times 1}$  will be denoted by  $T(\mathbf{p})$ , here the pose vector  $\mathbf{p} = [x, y, z, \boldsymbol{\theta}]$  contains position and orientation given in axis-angle representation  $\boldsymbol{\theta} = \theta\mathbf{e}$ .

The introduced map consistency losses can be used to learn parameters of the introduced depth correction models in a self-supervised way as follows. First, we build global map  $\mathcal{M}$  from point clouds  $\hat{\mathcal{X}}_k$  corrected by a depth correction model and transformed into a global reference frame using transformations  $\hat{T}_k$ :

$$\mathcal{M} = \bigcup_k \hat{T}_k \hat{\mathcal{X}}_k(w). \quad (9)$$

Each transformation  $\hat{T}_k = T_k \delta T(\mathbf{p}_k)$  is composed of initial transform  $T_k$ , provided as ground truth or from SLAM, and correction  $\delta T(\mathbf{p}_k)$  parameterized by vector  $\mathbf{p}_k \in \mathbb{R}^{6 \times 1}$ . Second, we define a set of points that are used during the optimization using the projection  $\phi: \mathcal{M} \rightarrow \mathcal{M}_\phi$  that selects a set of points that have *enough neighbors*, belong to *flat surfaces* and are observed from *diverse* (several different) *view-points*:

$$\mathcal{M}_\phi = \mathcal{M}_{nbr} \cap \mathcal{M}_{flat} \cap \mathcal{M}_{disp} \quad (10)$$

To put it formally, the individual subsets are defined as follows. The lower bound for the number of neighboring points is motivated by robust normal estimation.

$$\mathcal{M}_{nbr} = \{\mathbf{x}_i \in \mathcal{M} : \mathcal{N}_{\mathcal{M}}(\mathbf{x}_i) \geq N_{min}\} \quad (11)$$

As most of the points belonging to planar surfaces have a large incidence angle, we estimate the flat regions from the eigenvalue ratios.

$$\mathcal{M}_{flat} = \{\mathbf{x}_i \in \mathcal{M} : \frac{\lambda_1}{\lambda_2} \leq C_0, C_1 \leq \frac{\lambda_2}{\lambda_3} \leq C_2\} \quad (12)$$

Additionally, in order to evaluate map consistency information, we need to use points measured from multiple sensor poses.

$$\mathcal{M}_{disp} = \{\mathbf{x}_i \in \mathcal{M} : \sigma(\mathbf{p}(\mathbf{x}_i)) \geq \sigma_{min}, \sigma(p) = \text{tr} \Sigma_p\} \quad (13)$$

Here the  $\Sigma_p$  denotes view-points covariance matrix and the  $\sigma(\mathbf{p}(\mathbf{x}_i))$  is a view-point dispersion. An example of the individual subsets selected from a point cloud map is given in Figure 4.

Third, we compute mean consistency loss  $\ell$ , (7) or (8), over selected set of points  $\mathbf{x}_i \in \mathcal{M}_\phi$ ,

$$L(w, \mathbf{p}_1, \dots, \mathbf{p}_K) = \frac{1}{|\mathcal{M}_\phi|} \sum_{\mathbf{x}_i \in \mathcal{M}_\phi} \ell(\mathbf{x}_i), \quad (14)$$

from established neighborhoods  $\mathcal{N}_{\mathcal{M}}(\mathbf{x}_i, r)$  and respective sample covariance matrices  $Q_i$ . Forth, we compute the gradient of the loss with respect to the model parameters and individual transformation parameter vectors,  $\nabla_w L(w, \mathbf{p}_1, \dots, \mathbf{p}_K)$  and  $\nabla_{\mathbf{p}_k} L(w, \mathbf{p}_1, \dots, \mathbf{p}_K)$ , using backpropagation in the computational graph. Fifth, we update  $w$  and  $\mathbf{p}_k$  using a gradient descent algorithm. We repeat the above procedure for a fixed number of iterations and select the final model which minimizes the loss on a validation subset. The neighboring sets  $\mathcal{N}_{\mathcal{M}}(\mathbf{x}_i, r)$  computed for each point in global map,  $\mathbf{x}_i \in \mathcal{M}$ , are fixed during the optimization loop. Figure 2 describes map construction procedure, consistency loss computation, poses and depth correction model parameters updates (gradients back-propagation) from the Algorithm 1 which summarizes the described method.

---

#### Algorithm 1: Depth Correction Optimization

---

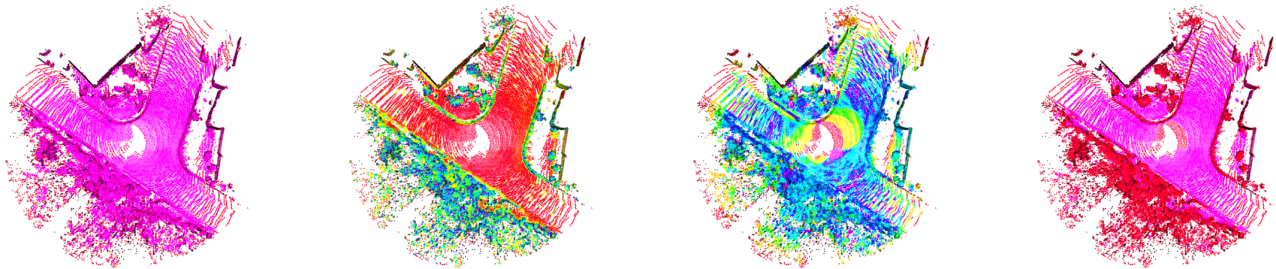
**Input:** Point sets  $\{\mathcal{X}_k\}_{k=1}^K$ , transformations  $\{T_k\}_{k=1}^K$ .

**Output:** Model parameters  $w$ .

- 1 Initialize model parameters  $w \leftarrow 0$ .
  - 2 Initialize transformation corrections  $\mathbf{p}_k \leftarrow 0$  for  $k = 1, \dots, K$ .
  - 3 **for**  $i = 1, \dots, N$  **do**
  - 4     Construct map according to (9),
  - 5      $\mathcal{M} \leftarrow \bigcup_k T_k \delta T_k(\mathbf{p}_k) \hat{\mathcal{X}}_k(w)$ .
  - 6     Select valid map elements according to (10),  
 $\mathcal{M}_\phi = \phi(\mathcal{M})$ .
  - 7     Compute loss according to (14),  
 $L(w, \mathbf{p}_1, \dots, \mathbf{p}_K) \leftarrow \frac{1}{|\mathcal{M}_\phi|} \sum_{\mathbf{x}_i \in \mathcal{M}_\phi} \ell(\mathbf{x}_i)$ .
  - 8     Compute loss gradients  $\nabla_w L(w, \mathbf{p}_1, \dots, \mathbf{p}_K)$  and  
 $\nabla_{\mathbf{p}_k} L(w, \mathbf{p}_1, \dots, \mathbf{p}_K)$ .
  - 9     Update  $w$  and  $\mathbf{p}_k$  using the computed gradients  
with a gradient descent algorithm.
  - 10 **Return**  $w$  minimizing validation loss. See text for the details.
- 

## IV. EXPERIMENTS AND RESULTS

We run a diverse set of tests with different criteria functions and depth correction models according to the scheme introduced in the Section III. A cross-validation method is used to split a data pull into training, validation and test sets. The parameters (models' coefficients and poses corrections) are learned on training part. The training performance is evaluated on validation data set. The best models according to map consistency criteria are selected and their performance on testing part is recorded. In the cross-validation scheme training, validation and testing data pulls are being swapped until all possible combinations are selected for the learning experiment. This algorithm is applied to the experiments described in first row of the Table I. The questions of to which extent depth correction models remove measurement bias and how it affects localization quality in SLAM scenarios are addressed next (second row of the Table I). In this part, we compare SLAM pipelines with and without usage of the trained depth correction models. In the table  $\mathbf{p}_i$  denotes



(a) Points with enough neighbors,  $\mathcal{M}_{nbr}$  (11),  $N_{min} = 10$  (b) Points belonging to flat surfaces,  $\mathcal{M}_{flat}$  (12),  $C_0 = 0.25$  (c) View-points dispersion,  $\mathcal{M}_{disp}$  (13),  $\sigma_{min} = 0.36$  (d) Filtering result,  $\mathcal{M}_\phi = \mathcal{M}_{nbr} \cap \mathcal{M}_{flat} \cap \mathcal{M}_{disp}$

Fig. 4. Filters applied to select from a set of scans the points used in the optimization. Example from the KITTI-360 data set [11]. (a) Outliers with not enough neighboring points are shown in red. (b) Red points have a smaller eigenvalues ratio,  $\frac{\lambda_1}{\lambda_2}$  (12) (c) The rainbow color gradient from red to violet describes the points in the map observed from different sensor poses (view-points). Blue and violet points are measured from higher number of view-points. (d) Resulting filtering mask, purple points are selected in the optimization, Algorithm 1.

TABLE I  
EXPERIMENTS OVERVIEW

| name                                | purpose   | estimated value  |
|-------------------------------------|---|--|
| model learning and poses correction | learning of depth-correction model with sensor poses refinement | $w, \delta \mathbf{p}_1, \dots, \delta \mathbf{p}_K$   |
| localization accuracy               | affect of depth correction models on localization accuracy      | $\sum_{i=1}^K \ \hat{\mathbf{p}}_i - \mathbf{p}_i\ _2$ |

ground-truth localization data, while  $\hat{\mathbf{p}}_i$  are the estimated poses, Section IV-D. Data containing sequences of lidar scans and their corresponding locations is utilized. We collect the data set in an indoor corridor environment, Section IV-B. This choice is motivated by the presence of large planar surfaces. It means that lidar beams fall at them at a wide spectrum of incidence angles which allows to observe the measurements bias.

#### A. Lidar Bias Estimation

The experimental setup to find out the dependence of lidar depth measurements on incidence angle is described in this section. We use the Ouster OS0-128 lidar<sup>3</sup>. A rigid  $0.50\text{m} \times 1.15\text{m}$  board with a metal frame is a target to measure with the lidar. The board could be rotated along the vertical axis Figure 6. The board position in space is measured both by the lidar and the theodolite Leica TS15. As the theodolite provides measurements of one point in space, we place the optical prism to be tracked by the Leica TS15 at the top left corner of the board. It provides the ground-truth location of the prism with 1mm accuracy. The position of the lidar relative to the board is also measured by the theodolite. In order to estimate the lidar orientation relative to the board's base (point of rotation), we run the ICP alignment of the point cloud sampled from the board mesh to the lidar point cloud at low angles of rotation of the board. The experimental board has

<sup>3</sup>hardware Rev C, <https://ouster.com/>

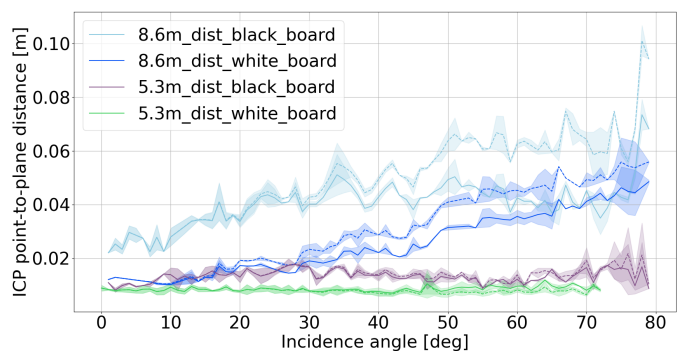


Fig. 5. Dependence of the point-to-plane distance between measured cloud (by Ouster OS0-128 lidar) and ground-truth plane (given by the calibration board surface) on incidence angle. Results are provided for experiments with the board placed at 5.3m and 8.6m distance from the lidar. Dashed lines correspond to experiments without bias removal, while solid ones represent the error with depth correction applied to the same point clouds. For one experiment its own color is assigned.

two different surfaces: without marking (white one), and the calibration side with AR-markers drawn on it (black one). We run experiments with the board being placed at 5.3m and 8.6m distances from the lidar and measure both of its sides. The goal is estimation of the difference between the measured board pose with respect to its ground truth for different orientations. Knowing the board's dimensions and the its base is fixed, the full pose of the board is given. As the metric, we calculate the point-to-plane distance between the board's measured point cloud and the plane representing the board's surface scanned by the lidar. The metric's values for different board angles are given in the Figure 5. The error achieves 10cm for incidence angles around  $80^\circ$  which confirms the results described in [9]. As dark surfaces have weaker reflective qualities, it was observed that the error is larger when the black side of the board was measured. The metric values for higher angles are not provided as there are almost no points measured by the lidar of the board oriented almost perpendicular. Note, that we did not observe dependence of the point-to-plane distance on incidence angles when the white-side target was placed reasonably close to the lidar (for 5.3m distance). Additionally,

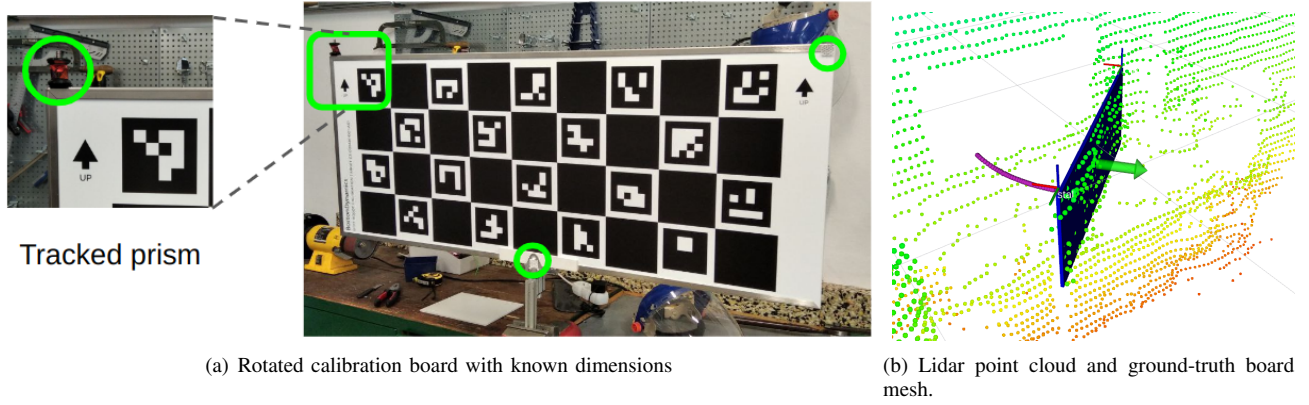


Fig. 6. Bias estimation experimental setup. *Left*: the rigid board which could be rotated around vertical axis. The optical prism is attached to its corner. The prism is tracked by the theodolite. Points marked with green circles are measured with the theodolite once before running the experiment in order to precisely estimate board’s dimensions. *Right*: tracked corner trajectory is visualized in purple. Colored points represent lidar point cloud. The ground-truth board location is visualized as a blue mesh with green arrow denoting its normal direction.

we provide the results corresponding to the point-to-plane value estimated for the lidar scans with the depth correction model (5) applied. They are depicted with the solid lines in the Figure 5. Due to dependence on depth the same model provides larger corrections for more distant measurements (8.6m).

### B. Data Set

In practice, however, it is not feasible to obtain exact positions and orientations of all the objects sensed by a lidar. In order to train the depth correction models using the self-supervised scheme described in Figure 2, we collect the following data set (called "FEE Corridor"). It contains point clouds captured in an office indoor environment with precise localization and ground-truth mapping information Figure 8. Two "stop-and-go" data sequences are provided of a robot with mounted Ouster OS0-128 lidar moving in the same environment. This data-capturing strategy allows recording lidar scans that do not suffer from an error caused by sensor movement. Individual scans from static robot positions are recorded. Additionally, point clouds recorded with the Leica BLK360 scanner are provided as mapping ground-truth data. The scanner is the tripod-mounted survey-grade lidar utilized to capture a detailed millimeter-accurate 3D-maps. Ground-truth 6 Degrees of Freedom (6 DoF) lidar poses are obtained using the SLAM introduced in [14] with alignment to the prerecorded ground-truth map. In addition, the lidar poses are recorded with the theodolite Leica TS15 tracker. The theodolite tracks a prism placed on the robot. In the corridor environment a direct line of sight of the prism is ensured for the whole time of recording. The tracker provides translation data ( $x_t$ ,  $y_t$ ,  $z_t$  coordinates of the prism relative to the tracker). The tracked poses are also included in the data set for reference. As we track only position of the prism, the orientation is provided by alignment of lidar scans to the pre-recorded with the Leica BLK360 scanner ground truth map.

As the importance of depth correction for autonomous cars was suggested already in the original work of Laconte

*et al.* [9], we also train and assess depth-correction models on the data collected in the driving scenarios, namely on a part of the KITTI-360 benchmark [11]. Here, the Velodyne HDL-64E sensor was used as point cloud measurements provider (at 10 fps), while ground-truth localization data was estimated with GPS/IMU data fused with visual features. We selected 8 sub-sequences each containing 50 consecutive scans (from 102 to 151) and corresponding lidar poses. From these sub-sequences for depth correction training and validation we take every 5th scan and its pose due to computational reasons and to ensure the sufficient overlap between the neighboring sensor locations. Sub-sequences of 500 consecutive (count from the beginning) scans were used in the evaluation of SLAM accuracy; see the Table III describing the sub-sequences from KITTI-360 used for depth correction training and evaluation.

### C. Model Learning and Poses Correction

In this experiment, the influence of depth correction models on map consistency is investigated. We divide FEE Corridor sequences into 8 sub-sequences. Each sub-sequence contains from 7 to 11 scans and corresponding lidar poses. Next train (4 sub-sequences), validation (2 sub-sequences) and test groups (2 sub-sequences) are selected from them. We utilize 4 cross-validation splits during training in order to have each sub-sequence in each group (train, validation or test). The model with the best performance on validation set is saved during training. Since ground-truth localization may be difficult to obtain due to the need for specialized hardware and usually a constrained stop-and-go operation, in addition to learning models’ parameters initial poses are being updated according to the optimization pipeline represented on Figure 2. Note, that lidar position and orientation correction could be preformed differently based on the type of localization error one would like to eliminate. In most of the real world cases, the sources of localization error are independent for individual poses at which laser scans were captured. Commonly, the transformations update would be computed as follow,  $\hat{T}_k = T_k \delta T_k$ , where  $k$  denotes time index, Algorithm 1. This poses update strategy was utilized in our work. However, if the error of localization

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

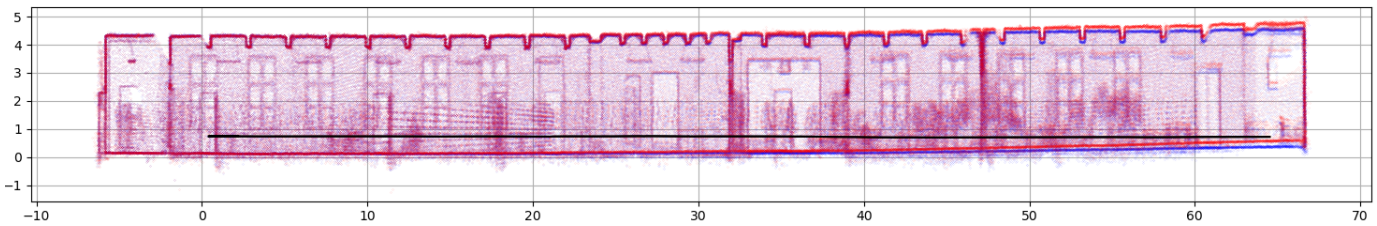


Fig. 7. Side view of the corridor environment mapped in the data set IV-B. Red point cloud denotes the map collected using baseline SLAM pipeline [14]. Blue map was constructed using depth correction model to remove bias from lidar measurements before using the scans in SLAM alignment procedure. The black line corresponds to robot trajectory tracked with the theodolite. The distance scales along vertical and horizontal axes are chosen differently for illustration purposes.

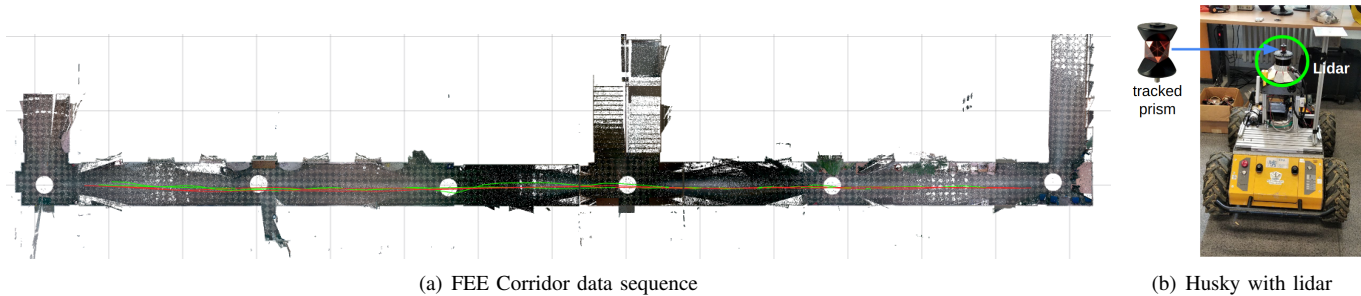


Fig. 8. Overview of the 3D model of the corridor (around 70m long) environment generated with a Leica BLK360 scanner and the ground-truth trajectory obtained by the proposed approach. One cell of the grid is equal to 5 meters. The two trajectories traveled by the robot (corresponding to collected data sequences) are shown in red and green respectively. Husky robot with mounted Ouster OSO-128 lidar and tracked prism is shown on the right.

TABLE II  
LOCALIZATION ACCURACY ON FEE CORRIDOR DATA SET.

| method                                | $\Delta\theta$ [deg]              | $\Delta t$ [m]  |
|---------------------------------------|-----------------------------------|-----------------|
| baseline, SLAM [14]                   | 1.49                              | 0.60            |
| SLAM [14] + DDD [16]                  | 7.26                              | 2.71            |
| ours, SLAM [14] + $\epsilon_p$ (4)    | <b><math>1.34 \pm 0.02</math></b> | $0.56 \pm 0.01$ |
| ours, SLAM [14] + $\epsilon_{sp}$ (5) | $1.36 \pm 0.02$                   | $0.56 \pm 0.01$ |

source could be considered negligible, it is still possible to encounter a bias connected with calibration inaccuracies (i.e. the error of a lidar location relative to the sensors suit origin). It is reasonable to assume that the pose correction term is constant and independent on time at which a laser scan was taken,  $\forall k = 1 \dots N : \delta T_k = \delta T$ .

The trained models are used further in order to visualize the effect of measurement bias suppression qualitatively. The point cloud map of the corridor environment is constructed with the help of the SLAM method [14] without depth correction. It is visualized in red on the Figure 7. The lidar's ground-truth trajectory (recorded with Leica TS15 tracker) on the side view is depicted in black. The map is being banded towards the ceiling as the distance from the corridor origin increases. We then run the same experiment with depth correction applied to raw lidar scans before being aligned in the SLAM algorithm. The resultant map is drawn in blue on the same Figure 7. With the lidar measurements correction, the map drifts less at the end of the corridor.

#### D. Localization Accuracy

TABLE III  
LOCALIZATION ACCURACY ON KITTI-360 DATA SET.

| method |            |          | SLAM [14]            |                | SLAM [14] + $\epsilon_p$ (4) |                |
|--------|------------|----------|----------------------|----------------|------------------------------|----------------|
| seq.   | start scan | end scan | $\Delta\theta$ [deg] | $\Delta t$ [m] | $\Delta\theta$ [deg]         | $\Delta t$ [m] |
| 00     | 1          | 500      | 1.95                 | 2.31           | <b>1.58</b>                  | <b>1.51</b>    |
| 03     | 1          | 500      | 4.23                 | 12.62          | <b>3.89</b>                  | <b>11.84</b>   |
| 04     | 1          | 500      | 2.05                 | <b>3.33</b>    | <b>1.66</b>                  | 3.39           |
| 05     | 1          | 500      | 1.85                 | 2.83           | <b>1.37</b>                  | <b>2.19</b>    |
| 06     | 1          | 500      | 2.21                 | 3.34           | <b>1.88</b>                  | <b>2.96</b>    |
| 07     | 1          | 500      | 7.61                 | 35.24          | <b>7.46</b>                  | <b>34.18</b>   |
| 09     | 1          | 500      | 4.04                 | 12.16          | <b>3.83</b>                  | <b>11.76</b>   |
| 10     | 1          | 500      | 3.75                 | 11.36          | <b>2.84</b>                  | <b>8.91</b>    |

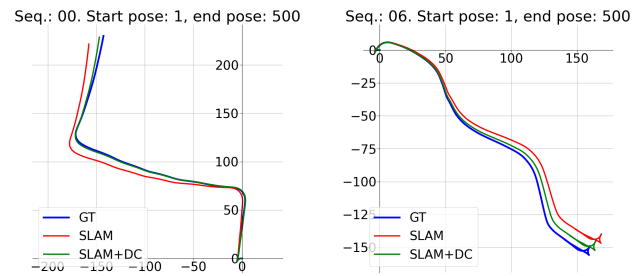


Fig. 9. Effect of depth correction on localization accuracy. The two examples from KITTI-360 data set are shown: the first 500 poses from the sequences 04 and 06 are given. Blue curves represent ground-truth trajectories, red ones are produced by the SLAM [14], the green ones correspond to our pipeline: SLAM [14] +  $\epsilon_p(4)$ . See Table III for more details.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

Up to this point we made sure, that the depth correction models help to create more accurate point cloud maps by elimination of the bias related to measuring objects with high incidence angle. The next reasonable question is, "Once we have more accurate lidar measurements, would it be possible to improve the localization accuracy of the sensor?". In order to address it, we benchmark the SLAM method, [14] on the sequences from the data set with the ground-truth localization information. In the first part of the experiment, the SLAM pipeline was running using the sensor measurements available in data sequences directly. Then we execute the same procedure with the difference: depth correction is applied to lidar scans before the SLAM point cloud alignment operation. Additionally, we run the similar experiment with the Deep Depth Denoising (DDD) method introduced in the [16] as the depth correction part. As the DDD model was trained with the RGB-D data, in the experiments point clouds are transformed to range images. The vertical resolution of an image is defined by the lidar's number of beams. The depth denoising operation is applied only for points in distance range (up to 6.5m) that was used by the authors of [16] during the model training. For each version of the experiment, the localization error is computed and averaged over time. We split the ground-truth trajectories from the FEE Corridor into poses uniformly spread 1m apart and record their corresponding time moments. We then evaluate the poses from SLAM at the same time moments and compare them to ground-truth poses. The results for depth correction models trained with minimum eigenvalue loss (7) are given in the Table II ( $\Delta\theta$  is orientation error,  $\Delta t$  is translation error). It is demonstrated that removing the bias from lidar scans improves localization accuracy by around 7%(4cm) in translation and 9%(0.15°) in orientation. For the DDD model experiments, we observe a significant localization error mostly caused by a drift in a vertical direction (more than 2.5m along  $z$ -axis only). We perform similar experiments on the KITTI-360 data set, refer to the Table III. It can be noticed, that the suppression of the depth bias from lidar measurements decreases both translation and orientation errors for all except one (number 04) sequences used in the experiments. Figure 9 contains examples of the SLAM pipeline runs on the first 500 scans from the sequences 00 and 06. The correction of lidar measurements takes around 80ms (including normals and incidence angles estimation) on a modern CPU. The implementation bottleneck lies in a point cloud normals estimation that could be overcome by GPU-based sparse tensor voting [12]. We run the SLAM evaluation experiments offline, however, the implementation speed-up is considered as a topic for the future work that will allow to use the module in real-time scenarios.

## V. CONCLUSION

In this paper, novel point cloud map consistency losses as well as depth measurement correction models were presented. To tackle the lack of ground-truth depth data, the models were trained using lidar scans taken at multiple viewpoints of the same scenes in a self-supervised manner. Once the models

were trained, it was shown that with their help the correction of lidar scans allows the construction of a higher-quality point cloud map. It was also demonstrated that depth correction reduces the drift of lidar localization. In addition to that, we release the data set with accurate ground-truth 3D-poses and mapping information for point cloud registration algorithms.

## REFERENCES

- [1] Nitin Agarwal, Sung-Eui Yoon, and M Gopi. Learning embedding of 3d models with quadric loss. In *30th British Machine Vision Conference (BMVC 2019)*, pages 1–11. British Machine Vision Association (BMVA), 2019.
- [2] E.P Baltasvias. Airborne laser scanning: basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):199–214, 1999.
- [3] Miguel A. Bandres and Julio C. Gutiérrez-Vega. Ince-gaussian beams. *Opt. Lett.*, 29(2):144–146, Jan 2004.
- [4] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.
- [5] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.
- [6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017.
- [7] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive lidar sensors. In *2019 IEEE intelligent vehicles symposium (IV)*, pages 1527–1534. IEEE, 2019.
- [8] Julius Kümmerle and Tilman Kühner. Unified intrinsic and extrinsic camera and lidar calibration under uncertainties. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6028–6034, May 2020.
- [9] Johann Laconte, Simon-Pierre Deschênes, Mathieu Labussière, and François Pomerleau. Lidar measurement bias estimation via return waveform modelling in a context of 3d mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8100–8106, May 2019.
- [10] Xiaolu Li and Yu Liang. Remote measurement of surface roughness, surface reflectance, and body reflectance with lidar. *Applied Optics*, 54(30):8904–8912, 2015.
- [11] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *Pattern Analysis and Machine Intelligence (PAMI)*, 2022.
- [12] Ming Liu, François Pomerleau, Francis Colas, and Roland Siegwart. Normal estimation for pointcloud using gpu based sparse tensor voting. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 91–96. IEEE, 2012.
- [13] W. Ni-Meister, D.L.B. Jupp, and R. Dubayah. Modeling lidar waveforms in heterogeneous and discrete canopies. *IEEE Transactions on Geoscience and Remote Sensing*, 39(9):1943–1958, Sep. 2001.
- [14] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.
- [15] A. V. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems V*, pages 435–442, Seattle, USA, June 2009.
- [16] Vladimiro Sterzentsenko, Leonidas Saroglou, Anargyros Chatzitofis, Spyridon Thermos, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. Self-supervised deep depth denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1242–1251, 2019.
- [17] G. Sun and K.J. Ranson. Modeling lidar returns from forest canopies. *IEEE Transactions on Geoscience and Remote Sensing*, 38(6):2617–2626, Nov 2000.
- [18] Orazio Svelto. *Principles of Lasers*. Springer, Boston, MA, 5 edition, 2010.
- [19] Michael A. Wulder, Joanne C. White, Ross F. Nelson, Erik Næsset, Hans Ole Ørka, Nicholas C. Coops, Thomas Hilker, Christopher W. Bater, and Terje Gobakken. Lidar sampling for large-area forest characterization: A review. *Remote Sensing of Environment*, 121:196–209, 2012.