

# Physics-based Simulation of Continuous-Wave LIDAR for Localization, Calibration and Tracking

Eric Heiden, Ziang Liu, Ragesh K. Ramachandran, Gaurav S. Sukhatme

**Abstract**—Light Detection and Ranging (LIDAR) sensors play an important role in the perception stack of autonomous robots, supplying mapping and localization pipelines with depth measurements of the environment. While their accuracy outperforms other types of depth sensors, such as stereo or time-of-flight cameras, the accurate modeling of LIDAR sensors requires laborious manual calibration that typically does not take into account the interaction of laser light with different surface types, incidence angles and other phenomena that significantly influence measurements. In this work, we introduce a physically plausible model of a 2D continuous-wave LIDAR that accounts for the surface-light interactions and simulates the measurement process in the Hokuyo URG-04LX LIDAR. Through automatic differentiation, we employ gradient-based optimization to estimate model parameters from real sensor measurements.

## I. INTRODUCTION

Light detection and ranging (LIDAR) sensors, also known as laser range finders (LRF), are active depth sensors that transmit and receive laser light to measure the distance to objects that reflect light back to the sensor. Since the advent of compact planar LIDARs, such as Sick LMS 200 or Hokuyo URG-04LX, they have become an important depth sensor in the perception stack of autonomous mobile robots.

LRF sensors exhibit unique noise characteristics. Their measurements are influenced by the distance to the object, the angle of incidence, the surface properties of objects (e.g. reflectivity and color), the environment temperature, and many other factors, which have been studied extensively [1], [2]. The goal of such characterizations is to derive an empirical sensor model “top-down”: given LIDAR measurements under varying conditions, such as surface types, inclination angles, etc., they produce a probabilistic model of noise characteristics for a particular sensor. This statistical model then further informs uncertainty-aware localization and mapping algorithms.

In this work, we take a different approach: starting from first principles on how laser light interacts with surfaces under various conditions, we implement a physically plausible simulation that recovers various effects encountered with physical LIDARs, such as spurious measurements, reflected and refracted rays. We simulate the measurement process of continuous-wave LIDARs where the phase shift between two amplitude-modulated laser light waves is calculated to measure the distance. Using automatic differentiation, we compute the gradients of all parameters in our simulation with respect to the simulated measurements and apply

The authors are with the Department of Computer Science, University of Southern California, Los Angeles, USA. heiden@usc.edu

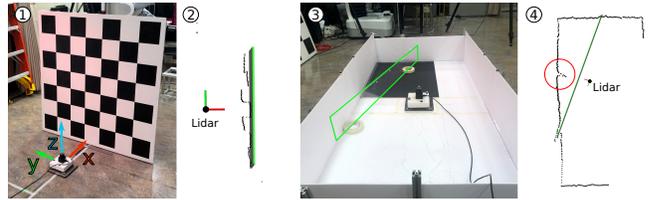


Fig. 1. Various effects encountered on the Hokuyo URG-04LX laser scanner. (1) LIDAR at a distance of 0.5 m in front of a checkerboard. (2) Measured point cloud obtained from the checkerboard with notable staircase pattern. (3) LIDAR in the cuboid environment, used for the localization and tracking experiments, with a sheet made of transparent plastic (outlined in green). (4) Sensed point cloud where the plastic sheet is invisible, except for a distortion that appears perpendicular to the LIDAR.

gradient-based optimizers to find the simulation parameters that most closely match the true observations.

To the best of our knowledge, our approach is the first physically plausible simulation that captures the measurement process of a continuous-wave LIDAR. Our real-world experiments show that our proposed model can accurately reproduce light-surface phenomena, such as refraction and specular reflection, that have not been considered in most previous LIDAR models.

## II. RELATED WORK

Detailed characterizations of 2D LIDAR sensors have been presented for the Hokuyo laser scanner URG-04LX in [3], [4], [2], [5]. The focus of these works is on experimentally analyzing how the LIDAR measurements are influenced by object color, material, incidence angle, etc. Our goal, in contrast, is to model the physics behind light-surface interaction and the LIDAR’s measurement process to allow for the automatic inference of the conditions that caused the observed measurements. A detailed analysis of the physical properties of LIDARs is given in [6], where the authors provide suggestions on how certain real-world effects can be modeled in simulation.

LIDARs and other active range finders are typically modeled in robot simulators to allow for the development of localization and mapping algorithms. Instead of accounting for the intricacies unique to these sensors – as done in the works on characterization – most general-purpose simulators for robotics, such as Gazebo [7] and MuJoCo [8], typically implement z-buffering or ray-casting to compute the depth and perturb it by zero-mean Gaussian noise, leading to a significant gap between simulation and reality [9]. In this work, we aim for a more detailed simulation that takes into

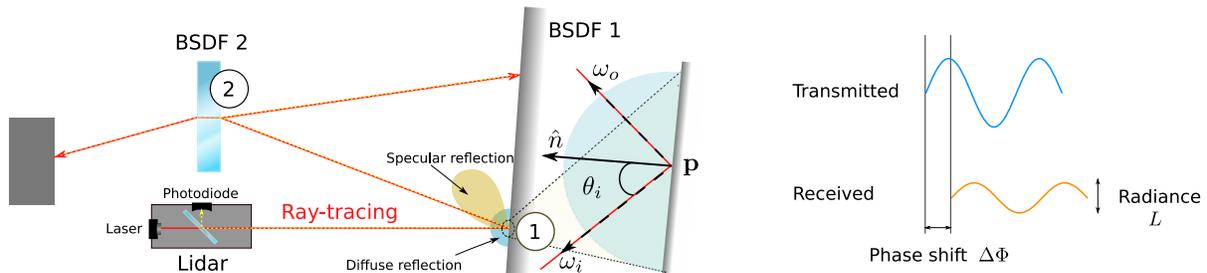


Fig. 2. *Left*: LIDAR principle where light emanating from a laser is captured by the photodiode on the other side of the half-transparent mirror. The laser ray is traced through the scene to compute the received radiance  $L$  (Equation 1) by considering various surface interactions through bidirectional scattering distribution functions (BSDF). BSDF (1) shows the uniform scattering of light due to blackbody radiation, and diffusive and specular reflection spread due to the microfacets in the region around the intersection point  $\mathbf{p}$  with surface normal  $\hat{n}$ . The magnified view of (1) depicts the vectors used for the formulation of BRDFs. BSDF (2) transmits and refracts a part of the light, while reflecting the other part (similar to glass). *Right*: attenuation in the LIDAR received wave compared to the transmitted wave. The phase shift of the received wave resulting from the light’s travel time from the laser to an object and back to the photodiode is used to compute the distance (cf. Equation 3).

account the interaction of laser light with different surfaces, and many other physical phenomena, as described in Sec. III. Similar to [10], we implement ray tracing to account for refractive and reflective surfaces. Additionally, we model the sampling process taking place in continuous-wave LRFs, as well as the detailed light-surface interaction that considers transmitting materials, such as glass and plastic.

Using gradients of models and algorithms has a long history in engineering, e.g. where gradients of ray-tracing equations have been derived to optimize the design of optical systems [11]. With the advent of automatic differentiation frameworks, accurate gradients can be computed algorithmically, powering model-predictive optimal control algorithms based on robotic simulators [12], [13], [14], design optimization [11], and various other use cases [15], [16]. In the context of simulating vision, prior work in computer graphics has focused on differentiable rendering systems [17], [18], [19], [20], [21] that tackle the problem of inverse graphics. Overall, our framework follows a similar idea of constructing a physics-based model of the real system for which the parameters can be efficiently estimated using gradient-based optimization.

### III. LIDAR MODEL

In the following section, we describe our simulation of a continuous-wave LIDAR, starting from the physics of light-surface interaction to the particular measurement process in the considered sensor.

#### A. Surface Interaction

When the laser light hits an object at a point  $\mathbf{p}$ , depending on the surface and angle of inclination, the beam is reflected, scattered and transmitted.

In computer graphics, researchers and practitioners use the rendering equation [22]

$$L_0(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{\Omega} f(\mathbf{p}, \omega_o, \omega_i) L_i(\mathbf{p}, \omega_i) |\cos \theta_i| d\omega_i \quad (1)$$

which expresses the radiance of the reflected light ray from point  $\mathbf{p}$  along the direction  $\omega_o$ , when a light ray of radiance

$L_i(\mathbf{p}, \omega_i)$  tracking the direction  $\omega_i$  collides with a medium at  $\mathbf{p}$ , integrated over all incoming light directions  $\omega_i$  from the unit hemisphere  $\Omega$ .  $\theta_i$  is the angle made by the incident light ray with the surface normal at  $\mathbf{p}$ . These quantities are depicted in Figure 2 (left).

The term  $L_e(\mathbf{p}, \omega_o)$  captures the effects of blackbody emissions from the material on which the light was incident. All substances having a temperature above absolute zero emit electromagnetic radiation. The radiance of such electromagnetic radiation with a wavelength  $\lambda$  emitted by a blackbody at a particular temperature can be derived using Planck’s law [22, Chapter 12.1.1].

The integral term in Equation 1 quantifies the amount of radiance imbibed by the reflected light ray along  $\omega_o$  from the light rays incident at  $\mathbf{p}$ . This term is referred to as a bidirectional reflectance distribution function (BRDF). Fresnel equations give the fraction of light reflected along  $\omega_o$ , denoted as  $F_r(\omega_o)$ . Thus, if we set the reflection distribution function  $f(\mathbf{p}, \omega_o, \omega_i) = F_r(\omega_o) \frac{\delta(\omega_i - \omega_o)}{|\cos \theta_i|}$ , then the integral reduces to  $F_r(\omega_o) L_i(\mathbf{p}, \omega_i)$ . Incidentally,  $F_r(\omega_o)$  is a function of the refractive indices of the participating media and can be computed in closed form [22, Chapter 8.2.1].

Apart from being specularly reflected, light can be scattered and diffused by the interacting surface. Diffuse reflection can be described using a BRDF (see Figure 2 for an illustration of uniform diffusion around a point, highlighted by the turquoise hemisphere). To accurately simulate the diffusive and reflective properties of surfaces, they are modelled as a collection of small microfacets. The distribution of microfacet orientations is described statistically to account for the roughness, specular and diffusive properties of surfaces.

Perfect diffusion is described by the Lambertian model that relates the intensity of the reflected light to the cosine of its incidence angle, resulting in surfaces (parameterized by brightness  $k_R$ ) that reflect light into all directions equally:

$$f(\mathbf{p}, \omega_o, \omega_i) = \frac{k_R}{\pi} \quad (2)$$

Besides perfect diffuse reflection, we additionally implement the Oren-Nayar model that represents the microfacet orientation angles by a Gaussian distribution which is parameterized

by its standard deviation. Common LIDAR models assume Lambertian surfaces everywhere. In practice, this assumption is often invalidated – particularly when mobile robots are employed in indoor areas, such as office spaces or shopping malls, where many highly reflective or transparent surfaces, such as windows, are present.

A bidirectional transmittance distribution function (BTDF) captures the amount of radiance carried by the light transmitted into the reflecting medium (BSDF(2) in Figure 2 shows light transmission in glass). Since  $F_r(\omega_o)$  is the fraction of light reflected,  $1 - F_r(\omega_o)$  gives the fraction of light transmitted or refracted. Therefore, by introducing Snell’s law of refraction to the integral term in Equation 1, a BTDF  $L_0^T$  can be written as

$$L_0^T(\mathbf{p}, \omega_r) = \frac{\eta_o^2}{\eta_i^2} (1 - F_r(\omega_o)) L_i(\mathbf{p}, \omega_i),$$

where  $\omega_r$  is the refracted direction of the light ray,  $\eta_o$  and  $\eta_i$  are the refractive indices of the participating media.

The combination of BTDFs and BRDFs describes the scattering of light upon its encounter with a medium. Any such combination is called a bidirectional scattering distribution function (BSDF). A material is often a combination of multiple BSDF – for example, transparent plastic, besides diffusing light slightly, has specular reflection and transmission components.

The integral in Equation 1 needs to be solved for each light ray which can in most cases only be approximated through sampling. Instead of sampling rays from the light source, we assume the LIDAR is the only source of radiance, and trace rays starting from the sensor’s location. Such technique is widely known as Whitted integration [23], a recursive ray-tracing algorithm that generates a tree of light rays (we use a maximum recursion depth of five). At each surface interaction, the raytracing tree is expanded by new rays that are cast into the scene according to the BSDF’s defined reflective and refractive properties.

In contrast to the rendering of a typical 2D image, we have to take into account the time it takes for the laser light to be reflected back so that we can simulate the measurement process in a LIDAR. Therefore, for each interaction at point  $\mathbf{p}$ , we record both the distance travelled so far along the render tree and compute the radiance received from  $\mathbf{p}$ . As the light travels a distance  $R$ , its intensity  $I$  attenuates according to the inverse square law  $I \propto \frac{1}{R^2}$ .

Based on the physics-based rendering toolkit `pbrt` [22], we implement ray-tracing with a watertight ray-triangle intersection algorithm and represent all scene geometry as triangle meshes. To efficiently find intersecting meshes, we store the scene geometry in a k-d tree.

### B. Measuring Distance

There are two main types of operating principles for LIDARs [24]: pulsed systems emit a laser pulse and measure the time until the pulse is received to compute the distance. Continuous-wave (CW) LIDARs emit a sinusoidal signal of known wavelength to estimate the distance by measuring the

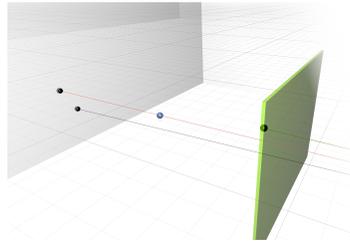


Fig. 3. Spurious measurement simulated by sub-sampling the laser beam using three rays (see Figure 5 II). Black spheres visualize endpoints of sub-samples, blue point is the final measurement, lying between the green obstacle closer to the LIDAR and the wall further from it.

phase difference between the transmitted and received signal (Figure 2 right). CW LIDARs typically require less powerful lasers, thus reducing hardware costs.

The LIDAR which we model in this work is the URG-04LX LRF by Hokuyo Automatic Co., Ltd. Being part of the URG series, as described in [25], it is a low-cost, planar CW LIDAR with a field-of-view of  $240^\circ$ , transmitting laser rays into 682 directions (thus achieving an angular resolution of  $0.352^\circ$ ) at a 10 Hz frequency. It modulates the amplitude of the signal to measure distance.

For a pulse ranging LIDAR, given the two-way travel time  $t_R$  of the emitted light, the measured range  $\hat{R}$  is computed by  $\hat{R} = t_R c / 2$ , where  $c$  is the speed of light. As we are interested in modeling CW LIDARs, the range is computed from the measured phase difference between the transmitted and received signal.  $\hat{R}$  is proportional to period time  $T = 1/f$  and phase difference  $\Delta\Phi$ :

$$\hat{R} = \frac{c}{2} \frac{T}{2\pi} \Delta\Phi = \frac{1}{4\pi} \frac{c}{f} \Delta\Phi. \quad (3)$$

In the case of the Hokuyo LRF, the amplitude-modulated laser light is received by an avalanche photodiode [25]. Throughout the ray-tracing process for a single measurement ray, we compute both the actual range  $R^*$  traveled up to the light-surface interaction plus the received radiance  $L$  (cf. subsection III-A). Each such return produces two new *received* waveforms over 15 periods at  $f_1 = 46.55$  MHz and  $f_2 = 53.2$  MHz frequencies with phase shifts  $\Delta\Phi^*(f_1)$ ,  $\Delta\Phi^*(f_2) + 90^\circ$ ,<sup>1</sup> and amplitude  $\propto L$  (Figure 2 right), respectively. For the two frequencies  $f_1$  and  $f_2$ , we sum up the received waveforms until the ray-tracing process is complete.

Next, the measurement process of the Hokuyo LIDAR begins by sampling both waveforms through 30 samples  $s_i$  at times  $t_i$ . The samples are reordered to form a high-resolution waveform (Figure 4 right). Using a simple discrete Fourier series approximation, the phase is computed via

$$\Phi = \arctan \left( \frac{\sum_i s_i \cos t_i}{\sum_i s_i \sin t_i} \right). \quad (4)$$

Analogously, the phase is computed for the transmitted signal and the phase difference to the received signal is computed

<sup>1</sup>Given actual range  $R^*$ , the actual phase shift for the waveform of frequency  $f$  is computed as  $\Delta\Phi^*(f) = 4\pi \frac{f}{c} R^*$ .

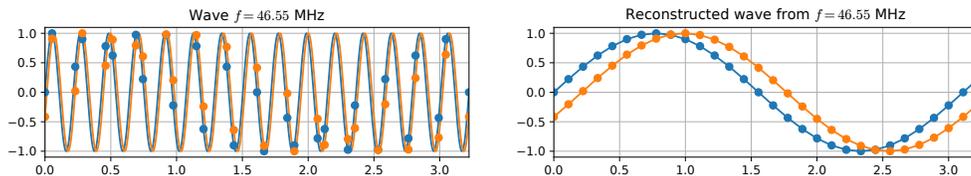


Fig. 4. Measurement process in the CW LIDAR URG-04LX [3]. Received (orange) and transmitted (blue) signal at a 46.55 MHz frequency with a phase shift  $\Delta\Phi$  resulting from a ranging distance of three meters. *Left*: analogue-to-digital conversion over 15 periods at 30 sampling steps  $s_i$  (dots) at times  $t_i$  (Equation 4). *Right*: higher-resolution waveform retained from the coarse samples  $s_i$  of the received and transmitted signals.

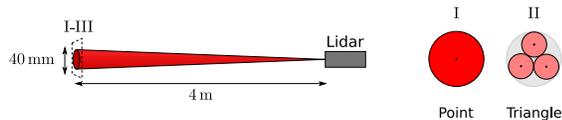


Fig. 5. *Left*: The beam divergence of the laser light emitted by the LIDAR amounts to a beam diameter of 4 cm at a distance of 4 m. *Right*: Cross-section views of approaches to sample the laser beam signals (cone) via rays. Simulating a LIDAR using a single ray (I) does not allow for capturing the beam divergence, while three (II) or more rays offer a more accurate simulation. For three beams, the contribution of each ray is weighted equally so that the total received radiance per beam is the same as for the point under the same conditions.

by subtracting both phases. From this phase shift  $\Delta\Phi$ , the measured distance is computed via Equation 3. The two phase shifts from the 46.55 MHz and 53.2 MHz waveforms are compared in order to resolve ambiguity in the phase difference of a single wave [25].

LIDARs exhibit spurious measurements (Figure 3), i.e., when the laser beam partially hits an obstacle closer to the scanner and another obstacle at a greater distance, the measured depth lies somewhere between these two obstacles. This phenomenon follows from the conical shape of the laser beam. As noted by Rosenberger et al. [6], a single infinitesimally thin ray is not sufficient to sample the behavior of beam divergence. Instead, we model the laser beam using three rays which are aligned around the central laser beam ray (Figure 5). At a distance of 4 m, the beam diameter is specified to be 40 mm.

#### IV. EXPERIMENTAL RESULTS

Besides retaining various effects from the real world, we demonstrate in our experiments the capability of our simulation to serve as a model that can be fitted to actual measurements from a physical LIDAR. To this end, we focus on scenarios where certain inputs to our model need to be estimated – such as the sensor’s pose, material properties of intersecting surfaces, and parameters internal to the sensing process – so that the resulting simulated measurements closely match the real data.

We investigate several scenarios and replicate them in our simulator to study various applications of the proposed LIDAR model. As shown in Figure 1 (3), we created a cuboid environment in which the LIDAR is placed. The environment has the dimensions (width  $\times$  depth  $\times$  wall height) 185 cm  $\times$  92 cm  $\times$  28 cm. We track the LRF using a Vicon motion capture system and place markers on the objects we want to track to achieve sub-millimeter groundtruth accuracy.

Throughout the experiments, the optimization objective is

$$\text{minimize}_{\theta} (\mathcal{L} = \|f(\theta) - y\|^2), \quad (5)$$

where the reality gap between simulated measurements  $f(\theta)$  and actual measurements  $y$  is to be minimized by adapting the simulation parameters  $\theta$ . These parameters are defined separately for each experiment.

Our simulator is implemented in C++ and uses the automatic differentiation framework Stan Math [26] to algorithmically compute gradients  $\frac{\partial \mathcal{L}}{\partial \theta}$ . Throughout all experiments, we use an off-the-shelf implementation of the gradient-based optimization algorithm L-BFGS with Wolfe line search from the Ceres library [27] to optimize Equation 5 with the calculated gradients throughout all experiments.

##### A. Localization in a Known Environment

Being placed in a simple cuboid environment where the wall geometries and surface properties are known, the goal in this experiment is to localize the LIDAR. Such a scenario is very common in LIDAR odometry where the current set of range measurements is registered with the point cloud obtained from a previous observation to obtain the rigid transform between the two poses.

Given a set of range measurements, we optimize Equation 5 for the sensor’s SE(2) pose  $\theta$ . The initial pose is defined as (0 m, 0 m,  $60^\circ$ ), the groundtruth pose is (0 m, 0 m,  $0^\circ$ ). We compare our approach to common point cloud registration algorithms. Given the depth information from the current pose (our initial guess), their task is to find the relative transform to a target point cloud, which stems from the actual LIDAR. Our baselines are Iterative Closest Point (ICP) [28], Generalized ICP (GICP) [29], and Normal Distributions Transform (NDT) [30]. We use their implementations from the Point Cloud Library [31].

As shown in Figure 6 (left), the optimization using our proposed model converges after 34 iterations to the correct pose. Executed on an Intel Core i7-8700K CPU (3.70 GHz), the computation takes 1.6 s, while the respective point cloud registration algorithms converge after 10-20 iterations (since we do not have access to the iteration-wise performance of these algorithms we show a flat line in the graph). GICP achieves the highest baseline accuracy with a final pose error of 0.13. Although the registration baselines finish the computation by one to two orders of magnitude faster than our approach, we note that our current implementation can be sped up considerably through parallelization since each of the 682 LRF returns can be computed independently.

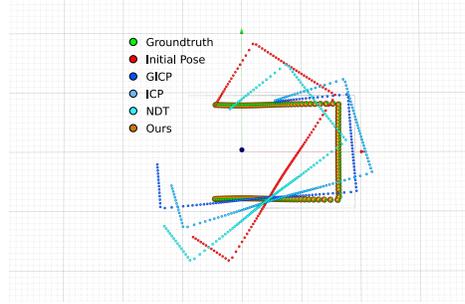
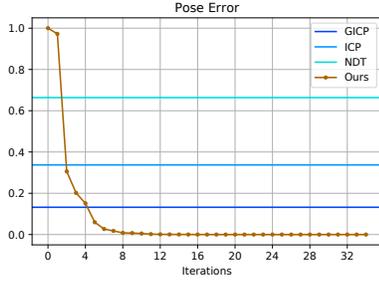


Fig. 6. *Left*: Evolution of the pose error, i.e., the SE(2) distance (accounting for yaw angle difference) between the estimated transform and the ground-truth pose of the LIDAR, while localizing the LIDAR in a known environment given its sensor output. The x-axis represents the number of iterations, no iteration-wise information was available for the point cloud registration algorithms GICP, ICP and NDT. *Right*: Visualization of the pose estimation results by transforming the point cloud of the given depth measurements by the estimated sensor transform.

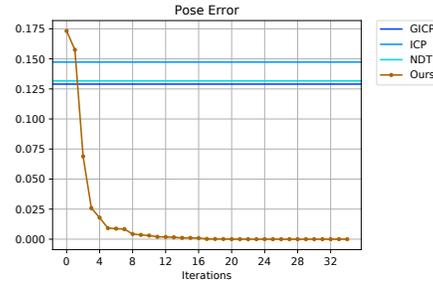
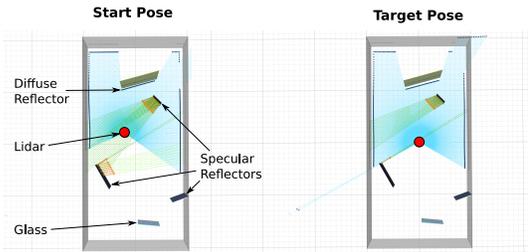


Fig. 7. *Left*: Experimental setup of localizing the LIDAR in a known, complex environment consisting of mirrors (specular reflectors), diffuse reflectors (Lambertian) and a glass element. The LIDAR (red) takes measurements (blue dots) at the start pose, and, given the measurements from the target pose, estimate the necessary transform. *Right*: Evolution of the pose error Equation 5 (SE(2) distance) between the estimated transform and the groundtruth.

While classical point cloud registration algorithms are generally able to find good solutions for simple scenarios such as the one shown in Figure 6, where the points follow a simple rectangular geometry, and the shape largely remains constant under the rigid transformation, depth measurements can become more complex when highly reflective surfaces (e.g., mirrors or glass surfaces) are present. Such phenomena become particularly critical in autonomous driving scenarios, where metallic car paint of darker colors absorbs the infrared light from the laser significantly [32]. Specular reflection and partial transparency in glass windows further distort the obtained depth measurements, affecting many optical sensor pipelines of robots deployed indoors.

To this end, we investigate an entirely simulated scenario where the LIDAR is exposed to multiple specular reflectors (mirrors), a glass surface and a diffuse surface (see Figure 7 left). Placed at the initial SE(2) pose (0.1 m, 0.1 m,  $10^\circ$ ), the ground truth pose (0 m, 0 m,  $0^\circ$ ) needs to be inferred given the sensor measurements. While the initial location is close to the actual pose, the point cloud registration algorithms achieve significantly less improvement over the initial guess compared to our optimization-based approach. As the point clouds differ significantly, a model that relies solely on their shapes is prone to find suboptimal solutions, while the gradient-based optimization applied to Equation 5 finds the accurate pose within 34 iterations (see Figure 7 right).

### B. Tracking a Mirror

Besides localizing the LIDAR, other objects in the environment can be tracked if they are modelled within the simulator. As shown in Figure 8, we place a mirror near the LIDAR and estimate its SE(2) pose using gradient-based optimization. Similar to subsection IV-A, our system identification approach applied to our model estimates the correct pose by minimizing the  $\ell_2$ -norm between the simulated and real depth measurements within six iterations (see Figure 8 right).

Our current approach suffers from the loss of gradient information when the object to be tracked is moved outside the field-of-view of the LIDAR, e.g. when it is behind the walls or in the blind spot of the sensor. Approaches from the computer graphics community, such as Soft Rasterizer [20], overcome the issue of missing gradient information by relaxing the intersection checks through a convex combination of blurred edges so that even faces that are hidden contribute to the intersection test. Future work is directed toward extending our model with such techniques to make tracking and localization more amenable to gradient-based optimization.

### C. Diode Calibration and Inferring Surface Properties

As noted in several characterizations of low-energy LIDARs, such as the URG-04LX [4], they suffer from a strong dependence of the measured depth on the received radiance. As shown in Figure 1 (2), the dark sections of the checkerboard appear closer than the more reflective white

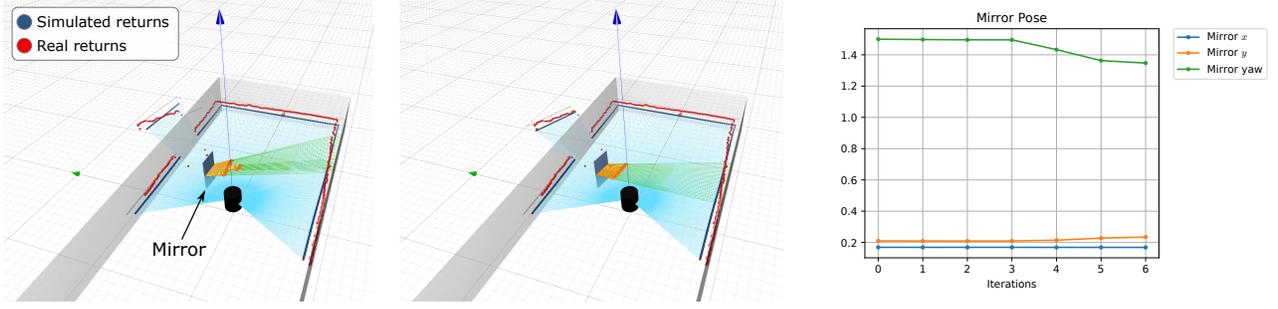


Fig. 8. *Left*: Initial setup for the tracking experiment where a mirror needs to be localized in a cuboid environment given real sensor measurements (red dots). Orange arrows with emanating green lines indicate reflection rays causing the measurement rays that hit the mirror to appear further, as the distance to the opposite wall is measured. *Center*: Converged pose estimate of the mirror after 6 iterations. *Right*: Evolution of the mirror’s SE(2) pose throughout the optimization.

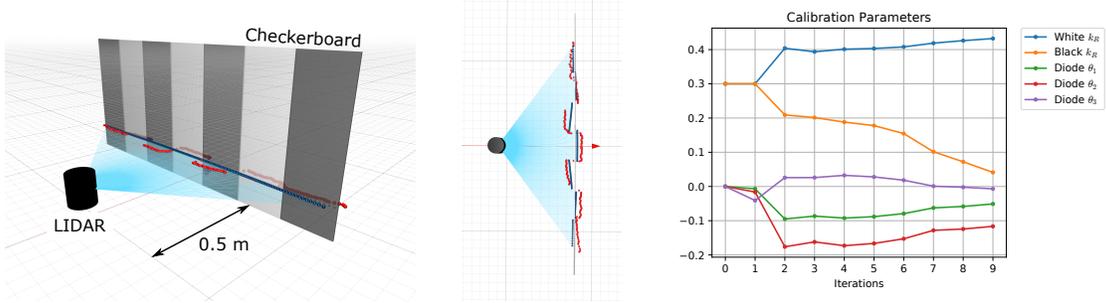


Fig. 9. *Left*: Experimental setup of the LIDAR in front of a checkerboard, where measurements are simulated (blue lines visualize rays, blue dots are placed at the simulated depth). Measurements not hitting the checkerboard are not shown. Red dots indicate the measurements from the actual Hokuyo scanner in front of a real checkerboard. *Center*: Top-down view of the simulation with converged surface parameters and diode calibration settings. The simulated measurements (blue) appear close to the real measurements (red). *Right*: Evolution of the surface properties (Lambertian reflectance coefficients  $k_R$  for the white and black sections) and the diode calibration parameters  $\theta_1, \theta_2, \theta_3$ .

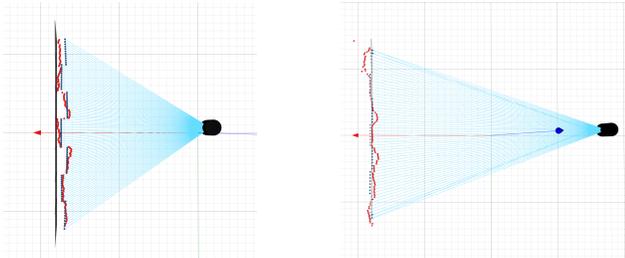


Fig. 10. Evaluation of surface properties and diode parameters obtained through the optimization described in subsection IV-C. Actual measurements from the real system are shown in red, simulated ranges in blue. *Left*: Checkerboard at a distance of 90 cm. *Right*: Checkerboard at a distance of 1.6 m.

sections, resulting in a staircase effect. While the exact cause of this effect is unknown to us, we seek to replicate this behavior through a data-driven approach and leverage our model to infer surface properties given the sensor’s range measurements.

We model the relationship between radiance and phase shift as a quadratic polynomial and regress its coefficients through optimizing Equation 5. This can be seen as a form of sum-of-squares optimization [33]. The phase shift  $\Delta\Phi$  to modulate the two continuous waves in Equation 3 is modified as follows:  $\Delta\Phi' \leftarrow \Delta\Phi - (aL^2 + bL + c)$ , where  $L$  is the received radiance and  $a, b, c \in \mathbb{R}$  are the polynomial coefficients to be estimated. We model the checkerboard using dark and bright stripes of the same dimensions as the real checkerboard (Figure 9 left) and assign Lambertian materials

of separate reflectivity coefficients  $k_R$  (cf. Equation 2) to the black and white sections.

It should be noted that this problem is particularly difficult since we do not have access to the actual intensities of each return but instead only the filtered distance measurements. Nonetheless, we are able to estimate the two different materials of a checkerboard, and converge to polynomial coefficients within 9 iterations (Figure 9 right) that closely match the real measurements (Figure 9 center). When we move the LIDAR from its initial configuration of 0.5 m distance to other distances in front of the checkerboard, we see that the simulated measurements still retain the staircase pattern from the real world accurately (Figure 10).

## V. CONCLUSION

We have presented a physics-based model of a continuous-wave LIDAR that allows for the optimization-based inference of model parameters. It captures the detailed interaction between the laser light and various surfaces, accounting for geometrical and material properties. Through experiments with real-world measurements from the Hokuyo URG-04LX LRF, we have demonstrated various applications of our model in localization, calibration and tracking.

Future research is directed towards extending our proposed simulation to model more powerful pulsed LIDAR sensors and investigate its application in real-world domains on a larger scale.

## REFERENCES

- [1] Cang Ye and J. Borenstein, "Characterization of a 2d laser scanner for mobile robot obstacle negotiation," in *IEEE International Conference on Robotics and Automation*, vol. 3, May 2002, pp. 2512–2518 vol.3.
- [2] L. Kneip, F. Tache, G. Caprari, and R. Siegwart, "Characterization of the compact hokuyo urg-04lx 2d laser range scanner," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1447–1454.
- [3] Hirohiko Kawata, Kohei Miyachi, Yoshitaka Hara, Akihisa Ohya, and Shin'ichi Yuta, "A method for estimation of lightness of objects with intensity data from SOKUIKI sensor," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Aug 2008, pp. 661–664.
- [4] Y. Okubo, C. Ye, and J. Borenstein, "Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation," in *Unmanned Systems Technology XI*, vol. 7332. International Society for Optics and Photonics, 2009, p. 733212.
- [5] R. Breda, "Experimental measurement of parameters of the spatial scanner Hokuyo URG-04LX," *Przegląd Elektrotechniczny*, vol. 88, no. 5b, pp. 132–135, 2012.
- [6] P. Rosenberger, M. Holder, M. Zirulnik, and H. Winner, "Analysis of real world sensor behavior for rising fidelity of physically based lidar sensor models," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 611–616.
- [7] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3.
- [8] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [9] P. J. Durst, C. Goodin, B. Q. Gates, C. L. Cummins, B. McKinley, J. D. Priddy, P. Rander, and B. Browning, "The need for high-fidelity robotics sensor models," *Journal of Robotics*, vol. 2011, 2011.
- [10] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "Blensor: Blender sensor simulation toolbox," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 199–208.
- [11] D. P. Feder, "Differentiation of ray-tracing equations with respect to construction parameters of rotationally symmetric optics," *J. Opt. Soc. Am.*, vol. 58, no. 11, pp. 1494–1505, Nov 1968.
- [12] M. Gifftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation," *Advanced Robotics*, vol. 31, no. 22, pp. 1225–1237, 2017.
- [13] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *Advances in Neural Information Processing Systems*, 2018, pp. 7178–7189.
- [14] E. Heiden, D. Millard, H. Zhang, and G. S. Sukhatme, "Interactive differentiable simulation," *CoRR*, vol. abs/1905.10706, 2019. [Online]. Available: <http://arxiv.org/abs/1905.10706>
- [15] J. F. M. Barthelemy and L. E. Hall, "Automatic differentiation as a tool in engineering design," *Structural optimization*, vol. 9, no. 2, pp. 76–82, Apr 1995. [Online]. Available: <https://doi.org/10.1007/BF01758823>
- [16] G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann, Eds., *Automatic Differentiation of Algorithms: From Simulation to Optimization*. New York, NY, USA: Springer-Verlag New York, Inc., 2002.
- [17] M. M. Loper and M. J. Black, "OpenDR: An approximate differentiable renderer," in *European Conference on Computer Vision*. Springer, 2014, pp. 154–169.
- [18] P. Henderson and V. Ferrari, "Learning to generate and reconstruct 3d meshes with only 2d supervision," in *British Machine Vision Conference (BMVC)*, 2018.
- [19] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, "Differentiable Monte Carlo ray tracing through edge sampling," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 222.
- [20] S. Liu, W. Chen, T. Li, and H. Li, "Soft Rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction," *arXiv preprint arXiv:1901.05567*, 2019.
- [21] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 38, no. 6, Nov. 2019.
- [22] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [23] T. Whitted, "An improved illumination model for shaded display," *Commun. ACM*, vol. 23, no. 6, pp. 343–349, June 1980. [Online]. Available: <http://doi.acm.org/10.1145/358876.358882>
- [24] J. Shan and C. K. Toth, *Topographic laser ranging and scanning: principles and processing*. CRC press, 2018.
- [25] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori, "Development of ultra-small lightweight optical range sensor system," in *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2005, pp. 1078–1083.
- [26] B. Carpenter, M. D. Hoffman, M. Brubaker, D. Lee, P. Li, and M. Betancourt, "The Stan math library: Reverse-mode automatic differentiation in C++," *arXiv preprint arXiv:1509.07164*, 2015.
- [27] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [28] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.
- [29] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics Science and Systems*, 2009.
- [30] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, Oct 2003, pp. 2743–2748.
- [31] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9–13 2011.
- [32] C. M. Seubert, "IR reflectivity of paint: Autonomy and CO<sub>2</sub> emissions," *Detroit Colour Council*, 2018.
- [33] P. A. Parrilo, *Semidefinite Optimization and Convex Algebraic Geometry*. SIAM, 2012.