PlatoNeRF: 3D Reconstruction in Plato's Cave via Single-View Two-Bounce Lidar

Tzofi Klinghoffer[1] Xiaoyu Xiang[*2] Siddharth Somasundaram[*1] Yuchen Fan[2]
Christian Richardt[2] Ramesh Raskar[1] Rakesh Ranjan[2]
[1]Massachusetts Institute of Technology [2]Meta

Project page: https://platonerf.github.io

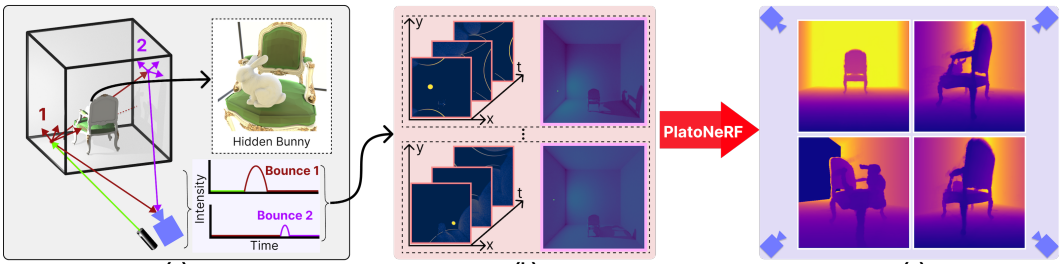


Figure 1. **PlatoNeRF**. We propose PlatoNeRF: a method to recover scene geometry from a single view using two-bounce signals captured by a single-photon lidar. **(a)** A laser illuminates a scene point, which diffusely reflects light in all directions. The reflected light illuminates the rest of the scene and casts shadows. Light that returns to the lidar sensor provides information about the visible scene, and cast shadows provide information about occluded portions of the scene. **(b)** The lidar sensor captures 3D time-of-flight images. **(c)** By aggregating several such images (by scanning the position of the laser), we are able to reconstruct the entire 3D scene geometry with volumetric rendering.

Abstract

3D reconstruction from a single-view is challenging because of the ambiguity from monocular cues and lack of information about occluded regions. Neural radiance fields (NeRF), while popular for view synthesis and 3D reconstruction, are typically reliant on multi-view images. Existing methods for single-view 3D reconstruction with NeRF rely on either data priors to hallucinate views of occluded regions, which may not be physically accurate, or shadows observed by RGB cameras, which are difficult to detect in ambient light and low albedo backgrounds. We propose using time-of-flight data captured by a single-photon avalanche diode to overcome these limitations. Our method models two-bounce optical paths with NeRF, using lidar transient data for supervision. By leveraging the advantages of both NeRF and two-bounce light measured by lidar, we demonstrate that we can reconstruct visible and occluded geometry without data priors or reliance on controlled ambient lighting or scene albedo. In addition, we demonstrate improved generalization under practical constraints on sensor spatial- and temporal-resolution. We believe our method is a promising direction as single-photon lidars become ubiquitous on consumer devices, such as phones, tablets, and headsets.

PlatoNeRF is named after the allegory of Plato's Cave, in which reality is discerned from shadows cast on a cave wall.
[*] Equal contribution.

1. Introduction

Recovering 3D scene geometry from a single-view is critical for many applications, ranging from autonomous vehicles (AV) to extended reality (XR). Consider playing a game of catch with a virtual ball in XR: if the ball drops and bounces behind your couch, it should bounce out in a physically realistic manner, dependent on the occluded geometry. Maintaining a complete and up-to-date scan of the scene is tedious for XR users and infeasible in many other applications, such as robotics and AV. Thus, methods are needed that recover geometry from single or few views; we address the former.

While neural radiance fields (NeRF) [26] are a popular representation for scene geometry, single-view 3D reconstruction with NeRF is challenging and remains an open problem. Existing methods in single-view 3D reconstruction with NeRF either rely on data priors [9, 21, 44, 48] or use visual cues, such as shadows, to infer occluded geometry from a single view [19, 20, 40, 46]. Approaches such as diffusion, generative adversarial networks, and transformers rely on data priors to exploit correlations between observations and a large corpus of training data. As a result, these methods are known to hallucinate content which, while statistically likely, may not be physically accurate. Other methods use shadows to infer occluded geometry when training NeRF [19, 20, 40, 46]. However, these methods struggle when the shadow is difficult to detect, such as in ambient light or low albedo backgrounds. In addition, these methods typically

predict relative depth, rather than absolute depth, which is important for many applications. To overcome these limitations, while still enabling physically-accurate reconstruction, we propose using two-bounce light measured with lidar.

Single-photon lidar systems, implemented with single-photon avalanche diodes (SPADs), offer an opportunity for accurate single-view 3D reconstruction. Lidar systems typically emit light into the scene and measure the time of flight (ToF) of the light to return to the sensor. As illustrated in Figure 1a, this light reflects off the scene multiple times — we refer to each reflection as a "bounce". While traditional lidar systems only exploit the first bounce of light from the scene back to the sensor, providing accurate absolute depth, recent work has shown that two-bounce time of flight, i.e. the time it takes for light to reflect off the scene two times before returning to the sensor, can enable reconstruction of occluded objects [7]. While promising, a limitation of existing methods is generalization to the lower spatial- and temporal-resolutions of lidars found on consumer devices.

Our method, called PlatoNeRF, addresses the limitations of single-view NeRF with two-bounce lidar and the limitations of two-bounce lidar with NeRF. The goal of PlatoNeRF is to reconstruct visible and occluded geometry from a single view using ToF measurements of two-bounce light from a single-photon lidar. Like Henley et al. [7], we illuminate individual points in the scene with a pulsed laser. Light is reflected off the illuminated points onto the rest of the scene before reflecting to the sensor. This light, referred to as two-bounce light, contains information about both scene depth and the presence of shadows created by the laser. Our experimental setup is described further in Section 3.1. Using the ToF measurements from multiple illumination points, we train NeRF to reconstruct the two-bounce ToF by modeling two-bounce optical paths. The presence or absence of two-bounce light reveals shadows, which allow occluded geometry to be inferred, and its ToF reveals depth. Our method is able to reconstruct 3D geometry with higher accuracy than existing single-view NeRF or lidar methods. Furthermore, using lidar allows our method to operate with higher ambient light and lower scene albedo than RGB methods that exploit shadows. We also demonstrate our method better generalizes to lower spatial- and temporal-resolutions than existing lidar methods due to our use of an implicit representation.

To summarize, our contributions are:

1. **Two-Bounce Lidar NeRF Model**: We propose a method to learn 3D geometry by modeling two-bounce light paths and supervising NeRF with lidar transients.
2. **Single-View 3D Reconstruction:** We demonstrate that our method is able to accurately reconstruct scenes from a single-view without hallucinating details of the scene.
3. **Analysis:** We study our method's robustness to ambient light, scene albedo, and spatial- and temporal-resolution.

In addition, we prepare a dataset of simulated scenes captured with a single-photon lidar. We use this data to evaluate our method and our baselines. Simulating such data is challenging and requires domain expertise. To lower the barrier to entry for machine learning with single-photon lidars and to drive future research in this direction, we will release this dataset upon publication.

Scope of this Work. Our work focuses on reconstruction of Lambertian scenes and we leave non-Lambertian scenes as future work. In addition, we focus on indoor scenes where there are multiple surfaces to reflect light. We assume laser scanning rather than flash illumination.

2. Related Work

Single-View Reconstruction. Single-view reconstruction is an ill-posed problem due to missing constraints. To address this, data-driven methods [11, 45, 47] hallucinate the invisible regions using learned 2D or 3D priors. Recently, inspired by the success of diffusion models in generation [8, 38], several methods explore distilling 3D correspondences from pretrained 2D text-to-image models [18, 32, 42, 43]. Others learn 3D priors to produce multi-view-consistent outputs conditioned on the input view [9, 21, 33]. While these methods can generate realistic images, they are unable to ensure physically accurate reconstruction of occluded regions without geometric cues, which is the focus of our work.

Neural Shape from Shadow. Shape from shadows (SfS) provides a physically-accurate way to infer occluded geometry based on the shadows it casts. While traditional methods use shadowgrams, space carving, and probablistic methods to infer SfS [17, 24, 34], in recent years, NeRF has been shown to be an effective representation for learning SfS [14, 19, 20, 40, 46]. These methods leverage volumetric rendering to reconstruct the object or scene based on the observation that pixels in shadow result from geometry between the shadowed point and the light source. However, the performance of these methods degrades when the shadow becomes invisible — either due to ambient light or low albedo backgrounds. In contrast, our method, while still relying on shadows to reconstruct occluded areas, is robust to these effects due to our use of lidar rather than RGB sensors.

3D Reconstruction with Single-Photon Lidars. Single-photon lidars record time-correlated light intensity and have been widely used for 3D reconstruction. We consider the most common type in our work: single-photon avalanche diodies (SPADs). SPAD-based methods often actively illuminate the scene and record the number of photons arriving at the sensor over time to infer scene geometry. Either visible or non-visible, e.g. near infrared, wavelengths of light can be emitted and detected. Each bounce of light in the scene reveals information about the scene's geometry. First-bounce light encodes scene depth [3, 13], while third-bounce light encodes partial information about the geometry of objects

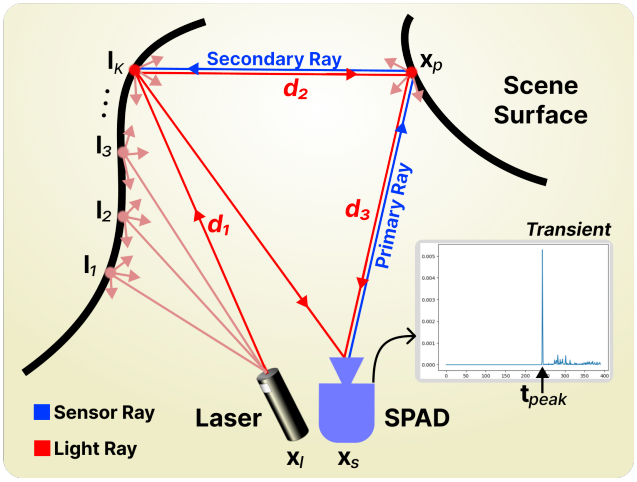


Figure 2. **Problem Definition.** We use a lidar system containing a SPAD at position $\mathbf{x}_s$ and a pulsed laser at position $\mathbf{x}_l$. The SPAD view is kept constant, while the laser sequentially illuminates different points in the scene, $\{\mathbf{l}_1, ..., \mathbf{l}_K\}$. For each illumination spot, we measure the time of flight for light to travel $\mathbf{x}_l \xrightarrow{d_1} \mathbf{l} \xrightarrow{d_2} \mathbf{x}_p \xrightarrow{d_3} \mathbf{x}_s$, shown by the captured *transient*.

that are outside the sensor's line of sight, e.g. around corners [16, 41]. NeRF has been used to exploit one- [1, 10, 23, 39] and three-bounce [5, 27, 35] light in lidar/ToF. We focus on two-bounce time of flight, which has recently been shown to encode the geometry of occluded objects [6, 7, 37]. Henley et al. [7] propose a two-step approach, first estimating scene depth from two-bounce returns, and then occluded geometry based on shadows inferred from the presence or absence of two-bounce returns. Inspired by this work, we propose a single, unified pipeline with NeRF to reconstruct both properties. Our work has three main benefits over [7]: (1) a unified approach for both visible and hidden geometry, (2) smoother scene reconstruction, and (3) better generalization to lower spatial (e.g. 32×32) and temporal resolution regimes, which are key limitations on consumer devices [25].

3. NeRF from Single-View Two-Bounce Lidar

In this section, we outline a method to extract 3D geometry from two-bounce transient measurements with NeRF. In Section 3.1, we describe the experimental setup, image formation model, and two-bounce transients. In Section 3.2, we describe how NeRF is trained with supervision from two-bounce transients. In Section 3.3, we provide implementation details that enable replication of our method and results.

3.1. Notations and Problem Definition

Experimental Setup. Our experimental setup is shown in Figure 2. The lidar system consists of a SPAD sensor and pulsed laser at known positions $\mathbf{x}_s$ and $\mathbf{x}_l$ respectively. The laser sequentially points at K different points $\mathcal{A} = \{\mathbf{l}_1, ..., \mathbf{l}_K\}$. For each illumination point $\mathbf{l}_k$, an image $\mathbf{i}_k$ is captured, resulting in a set of K image captures $\mathcal{I} = \{\mathbf{i}_1, ..., \mathbf{i}_K\}$, as illustrated in Figure 1.

One-Bounce vs. Two-Bounce Light. SPAD sensors are able to infer properties of a scene by measuring light that has interacted with the scene. In this problem, we are interested in inferring 3D scene geometry from *one-bounce* and *two-bounce* light, where "bounce" denotes the number of times light reflects off a scene surface. In Figure 2, light that travels along the path $\mathbf{x}_l \rightarrow \mathbf{l} \rightarrow \mathbf{x}_s$ is one-bounce light because it undergoes one reflection at $\mathbf{l}$. Similarly, light that travels along the path $\mathbf{x}_l \rightarrow \mathbf{l} \rightarrow \mathbf{x}_p \rightarrow \mathbf{x}_s$ is referred to as two-bounce light because it undergoes two reflections at $\mathbf{l}$ and $\mathbf{x}_p$. We refer to each illumination point $\mathbf{l}$ as a *virtual source* because it acts as a point light source. Similarly, we define $\mathbf{x}_p$ as a *virtual detector* because it refers to the scene point that is observing light from $\mathbf{l}$. In measurement $\mathbf{i}_k$, the pixel observing scene point $\mathbf{l}_k$ measures one-bounce signal, and all other pixels (e.g. $\mathbf{x}_p$) measure two-bounce signals or shadows. In general, one-bounce light arrives at the sensor earlier in time because it travels a shorter optical pathlength. One-bounce light also generally has higher intensity because light intensity is attenuated after every surface reflection.

Transient Measurement. Each image $\mathbf{i} \in \mathbb{R}^{N_u \times N_v \times N_t}$ is a *transient* measurement, where N_u and N_v represent the spatial resolution and N_t denotes the number of timing bins. A transient $\mathbf{i}(u, v, t)$ measures the amount of light arriving at every pixel (u, v) at a given time t. A pixel measurement $\mathbf{i}(u, v, :)$ measures a histogram of light intensity as a function of time. Each bin of the histogram is discretized to a timing resolution, or bin width, of t_{res}. For our experiments, we choose $t_{\text{res}} = 128$ ps, meaning that we can resolve the pathlength of light up to a precision of 3.8 cm. An example histogram two-bounce signal is plotted in Figure 2. The location of the two-bounce peak t_{peak} in the histogram is directly correlated to the pathlength d that the light travels via the following equation

$$t_{\text{peak}} = \frac{d}{c} = \frac{d_1 + d_2 + d_3}{c} \tag{1}$$

$$= \frac{\|\mathbf{x}_l - \mathbf{l}\|_2 + \|\mathbf{l} - \mathbf{x}_p\|_2 + \|\mathbf{x}_p - \mathbf{x}_s\|_2}{c}, \tag{2}$$

where $c \approx 3 \cdot 10^8$ m/s is the speed of light, d_1 corresponds to the distance between the laser $\mathbf{x}_l$ and the virtual source $\mathbf{l}$, d_2 is the distance between the virtual source $\mathbf{l}$ and virtual detector $\mathbf{x}_p$, and d_3 is the distance between the virtual detector $\mathbf{x}_p$ and sensor $\mathbf{x}_s$, as shown in Figure 2.

Shadow Measurement. Equation 2 assumes that a direct path exists between $\mathbf{x}_p$ and $\mathbf{l}$. However, if $\mathbf{x}_p$ lies in shadow, no two-bounce signal will be measured (i.e., no pulse will be observed). $\mathbf{x}_p$ is defined to lie in shadow if an opaque object lies along the ray connecting $\mathbf{l}$ and $\mathbf{x}_p$. Because $\mathbf{l}$ is modeled as a point light source, we neglect any diffraction effects and soft shadows that are common with area sources.

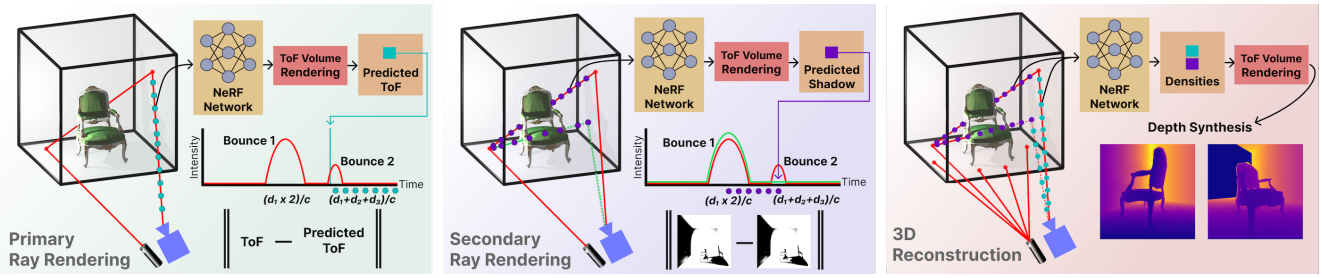


Figure 3. **Method.** PlatoNeRF learns 3D scene geometry from single-view two-bounce lidar time of flight, modeled with NeRF. Our method consists of three steps. **(a)** First, we render *primary rays* from the camera to the scene (Section 3.2.1). **(b)** Second, we model rays that scatter and travel to the virtual light (the point where light rays first hit the scene) (Section 3.2.2). Both steps are supervised with transients measured by a single-photon lidar. **(c)** Third, we find that reconstructing the two-bounce time of flight enables 3D reconstruction (Section 3.2.3).

Problem Statement. The resulting transient measurements will contain information about one-bounce signals, two-bounce signals, and shadows. The one-bounce and two-bounce signals provide information about objects that are visible to the sensor, and the shadows provide information about occluded portions of the scene. Using these measurements, we will reconstruct the 3D geometry of visible and occluded portions of the scene. Note that although we capture N measurements, the measurements are captured from the same view with only the laser being scanned.

3.2. Two-Bounce Volumetric Lidar Rendering

We parameterize our scene as a neural radiance field (NeRF). The MLP $f_\theta : \mathbb{R}^3 \to \mathbb{R}$ predicts a volume density σ for every input 3D scene point $\mathbf{x} = (x, y, z)$. The 3D geometry of the scene can then directly be estimated from $\sigma(\mathbf{x})$. The goal of this subsection is to synthesize transient images by developing a renderer that can map densities σ to predicted transients $\hat{\mathbf{i}}$. These synthesized transient measurements can then be used to train the NeRF in an analysis-by-synthesis framework. Note that, unlike a vanilla NeRF, we are not computing radiance, because we only reconstruct 3D geometry, not texture. Our method is summarized in Figure 3.

To render two-bounce transients, we must render along two types of rays: (1) *primary rays* and (2) *secondary rays*. Primary rays are defined as $\mathbf{r}_p(\lambda) = \mathbf{o}_p + \lambda \mathbf{d}_p$, where $\mathbf{o}_p = \mathbf{x}_s$ and $\mathbf{d}_p$ is determined by the camera matrix. Secondary rays are defined as $\mathbf{r}_s(\lambda) = \mathbf{o}_s + \lambda \mathbf{d}_s$, where $\mathbf{o}_s = \mathbf{x}_p$ and $\mathbf{d}_s = (\mathbf{l} - \mathbf{x}_p)/|\mathbf{l} - \mathbf{x}_p|$. We assume that the position of $\mathbf{l}$ is known by using standard time-of-flight techniques [4]. Consistent with NeRF literature, all equations are expressed with respect to a single pixel measurement.

3.2.1 Rendering Primary Rays

The goal of rendering along the primary ray is to compute the two-bounce time-of-flight $t_{\text{peak}} = d/c$ by determining the depth d_3 of $\mathbf{x}_p$. Once the location of $\mathbf{x}_p$ is known, d_1 and d_2 can subsequently be computed because $\mathbf{l}$, $\mathbf{x}_s$, and $\mathbf{x}_l$ are already known (Equation 2). First, the MLP is queried at P sampled points along the primary ray $\mathbf{r}_p$ between the near plane and far plane to output densities $\sigma_1, ..., \sigma_P$. The depth along the ray can be computed from the densities as

$$\hat{d}_3(\mathbf{r}_p) = \sum_{i=1}^{N} T_i \alpha_i t_i \tag{3}$$

$$\text{where } T_i = \prod_{j=1}^{i-1} \left(1 - \alpha_j\right) \text{ and } \alpha_i = \left(1 - e^{-\sigma_i \delta_i}\right), \tag{4}$$

where $\delta_i = t_i - t_{i-1}$ is the distance between two samples along a ray. This equation can be interpreted as a discretized expectation integral, where the product $T_i \alpha_i$ is the probability that the ray terminates *exactly* at t_i (i.e. $d_3 = t_i$). T_i is the transmittance a distance t_i along the ray and models the probability that the ray is not terminated before arriving at t_i. α_i denotes the probability of the ray terminating at t_i and is commonly used in graphics for alpha compositing.

3.2.2 Rendering Secondary Rays

The goal of rendering secondary rays is to determine if $\mathbf{x}_p$ lies in shadow or not. Every primary ray has a corresponding secondary ray, which is determined by (1) computing the depth d_3 along the primary ray and (2) connecting the estimated $\mathbf{x}_p$ to the virtual light source $\mathbf{l}$. The secondary ray connects the virtual source $\mathbf{l}$ and the virtual detector $\mathbf{x}_p$. Intuitively, if $\mathbf{x}_p$ lies in shadow, density along the secondary ray will be high. Otherwise, the density will remain low. The probability that $\mathbf{x}_p$ does *not* lie in shadow is

$$p_{\text{shadow}} = \prod_{j=1}^{N-1} \left(1 - \alpha_j\right). \tag{5}$$

The product integral is effectively the transmittance along the secondary ray, where a low transmittance indicates $\mathbf{x}_p$ lies in shadow. Note that, unlike the primary rays, the near and far planes of the secondary rays are known; the near plane is defined as $\lambda_n = 0$ and the far plane as $\lambda_f = d_2$, enabling these rays to be rendered more efficiently.

3.2.3 Synthesizing Transient Measurements

Using the synthesized two-bounce time-of-flight and shadow measurements, we can compute a loss based on the input transient measurement. We assume that $\mathbf{l}$, $\mathbf{x}_s$, and $\mathbf{x}_l$ are known. In addition, we assume that we have a binary mask $\mathbf{m}_k \in \mathbb{R}^{N_u \times N_v}$ for each measurement $\mathbf{i}_k$ that segments the transient image into shadowed and unshadowed pixels. Details on how to compute these quantities using the raw transient measurements are further explained in Section 3.3.

Distance Loss. The distance loss measures the accuracy of the synthesized two-bounce time of flight. Recall that rendering the primary ray enables estimation of the two-bounce time of flight using Equation 2 and $\mathbf{x}_p$ obtained from Equation 3. The distance loss is expressed as

$$\mathcal{L}_{\text{primary}} = \|t_{\text{peak}} - \hat{t}_{\text{peak}}\|_2^2, \tag{6}$$

where t_{peak} is the time of flight observed in the transient measurement, and $\hat{t}_{\text{peak}}$ is the two-bounce time-of-flight predicted by the NeRF rendering algorithm. Note that the distance loss is only computed on unshadowed pixels because a two-bounce signal will not exist for a shadowed pixel. However, we would still be able to estimate scene depth for these shadowed pixels because it is unlikely that the pixel will be shadowed in all N images of $\mathcal{I}$ due to illumination diversity.

Shadow Loss. The shadow loss determines if $\mathbf{x}_p$ is correctly classified as a shadowed or unshadowed pixel based on the rendered value p_{shadow}. The shadow loss is computed using the output rendering from the secondary ray in Equation 5. The shadow loss is expressed as

$$\mathcal{L}_{\text{secondary}} = \|s - \hat{p}_{\text{shadow}}\|_2^2, \tag{7}$$

where $s \in \{0, 1\}$ is a binary value from $\mathbf{m}_k$ indicating whether the transient measurement observed a shadow at the pixel. Unlike the distance loss, the shadow loss is computed for all pixels, shadowed and unshadowed.

Combined Loss Function. The final loss function can be expressed as a weighted sum of the distance and shadow loss

$$\mathcal{L} = \mathcal{L}_{\text{primary}} + \beta \mathcal{L}_{\text{secondary}}, \tag{8}$$

where β is a hyperparameter. Once the MLP is trained on this loss function, the predicted volume density can be extracted by densely sampling 3D scene points and querying the MLP at these points. The resulting densities can be used to render a depth map from any viewpoint or to generate a 3D mesh with marching cubes [22]. The loss, while simple in form, enables reconstruction of both the visible and occluded scene using only physically-based measurements, without data priors.

3.3. Implementation Details

Data Pre-Processing. Our method requires five inputs for each pixel: (1) the sensor location $\mathbf{o}_p = \mathbf{x}_s$ and ray direction $\mathbf{d}_p$, (2) laser location $\mathbf{x}_l$, (3) the distance from the laser to the virtual source $\|\mathbf{l} - \mathbf{x}_l\|$, (4) two-bounce time of flight t_{peak}, and (5) if the pixel is in shadow or not. (1) can be computed using camera matrices and (2) is assumed to be calibrated. We use signal processing to extract (3–5). We compute time-of-flight by using a template of the laser pulse shape as a match filter, and compute the cross-correlation of the match filter with the histogram at every pixel. The peak of the cross-correlation yields the time-of-flight at that pixel, yielding (4), and the maximum value of the cross-correlation yields the confidence that a pulse was measured. We filter out one-bounce returns by finding rays that are parallel or close to parallel to $\mathbf{l}$, and setting the corresponding histogram intensities to zero. Then, (5) can be computed by thresholding the confidence map to yield the binary shadow mask. (3) can be computed by determining the pixel with one-bounce return and computing the distance as $d_1 = ct_{1B}/2$, where t_{1B} is one-bounce time-of-flight, since the laser and sensor are roughly co-located in our experiments.

Training. For the first $25{,}000$ iterations of training, β is set to 0. After $25{,}000$ iterations, when an accurate initial estimate of the virtual detector $\mathbf{x}_p$ is obtained, we set β to $1/6{,}000$ in most experiments to encourage $\mathcal{L}_{\text{primary}}$ to continue to improve after activating the shadow loss. The NeRF MLP is queried twice for each primary ray, once with coarse samples and once with fine samples, followed by coarse and fine sampling along the secondary rays.

Implementation. PlatoNeRF is implemented in PyTorch [28] and is built off of vanilla NeRF [26]. As in NeRF, we use the Adam optimizer [15] and set an initial learning rate of 5×10^{-4}, which decays exponentially over training.

4. Experiments

We validate our method on the task of 3D reconstruction across several scenes. First, we introduce the simulated datasets that we make available to accelerate future work in learning-based methods for single-photon lidars. Then, we share our results, comparisons, and ablations on spatial and temporal resolution, ambient light, low-albedo backgrounds, non-planar backgrounds, and number of illumination points.

4.1. Datasets

We validate our method on four simulated datasets and a real dataset, each described below.

Simulated Datasets. We create datasets of four scenes of a room with either a chair, bunny, dragon, or occluded bunny in a chair, shown in Figure 4. The scenes are captured using a time-of-flight extension of Mitsuba [12] created by Pediredla

Table 1. **Depth evaluation**. We compare PlatoNeRF to both lidar- and RGB-based single-view 3D reconstruction methods, BF Lidar [7] and S^3-NeRF [46], respectively. Depth metrics are reported (in m for L1 and dB for PSNR) both for the train view and 120 novel test views.

	Chair Scene			Dragon Scene			Bunny Scene			Occlusion Scene		
	Train View	Test Views		Train View	Test Views		Train View	Test Views		Train View	Test Views	
Approach	L1 Depth ↓	L1 Depth ↓	PSNR ↑	L1 Depth ↓	L1 Depth ↓	PSNR ↑	L1 Depth ↓	L1 Depth ↓	PSNR ↑	L1 Depth ↓	L1 Depth ↓	PSNR ↑
BF Lidar	0.0348	0.1837	19.63	0.0233	0.1049	22.58	0.0339	0.0660	25.16	0.0341	0.2151	18.96
S^3–NeRF	0.0602	0.1178	22.80	0.0619	0.1042	25.06	0.0633	0.0877	27.67	0.0682	0.1336	22.51
PlatoNeRF	**0.0222**	**0.0862**	**26.58**	**0.0186**	**0.0870**	**28.45**	**0.0191**	**0.0601**	**30.26**	**0.0185**	**0.0836**	**27.33**

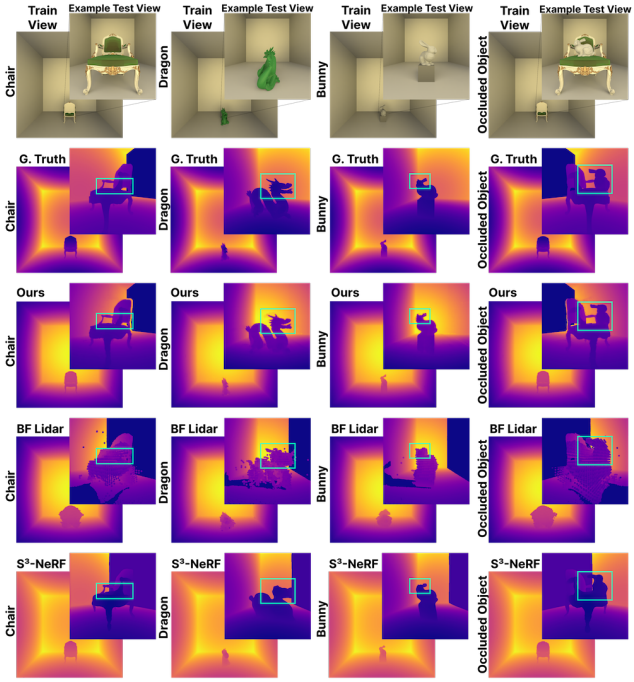


Figure 4. **Qualitative Depth Results**. We provide qualitative results for predicted depth on both train and novel test views, comparing our method, BF Lidar [7], and S^3–NeRF [46] to the ground truth across four scenes. Each method is trained from the one train view shown and reconstructs the entire scene.

et al. [29]. For each scene, we heuristically choose $N = 16$ points in the left and right parts of the scene, corresponding to the left and right walls, to illuminate. For each point that is illuminated, we record a transient image. Our scene is measured using a 512×512 SPAD with a temporal resolution of 128 ps (3.84 cm). Intensity is measured for 0.05 μs (15 m) per illumination spot, resulting in 391 timing bins per pixel. We also render corresponding ground truth depth images both from the training view and across 120 test views around the scene for evaluation. Transient data, depth, and all sensor and illumination parameters will be released.

Real Dataset. We use a dataset of single-photon lidar data captured by Henley et al. [7] to validate our method outside of simulation. The dataset captures a simple indoor scene, shown in Figure 5, containing a mannequin and box. The scene is captured with a 200×200 pixel sensor with an instrument response function of 128 ps (full width at half maximum). The scene is illuminated with 16 laser spots and a per-pixel transient is captured for each laser spot.

Table 2. **Point Cloud Evaluation**. We compute the Chamfer distance between the point clouds generated by each method. Metrics are averaged over all four simulated scenes and std is reported.

Approach	Chamfer (Mean)↓	Std.↓
BF Lidar	0.0465	0.0014
S^3-NeRF	0.4129	0.0021
PlatoNeRF	**0.0280**	**0.0014**

4.2. Results

Baselines. We compare our work with two methods, one that uses two-bounce lidar for single-view 3D reconstruction without learning and one that uses shadows measured by an RGB camera to train NeRF. We note that, to the best of our knowledge, we are the first to model two-bounce lidar with NeRF and so there are not direct comparisons for this task.

1. **Bounce-Flash Lidar**: Our work is inspired by Bounce-Flash (BF) Lidar [7], which models two-bounce lidar analytically to estimate visible depth and occluded geometry from a single view, using geometric constraints and shadow carving [34], respectively. BF Lidar's output is one point cloud (PC) for visible and one for occluded geometry, which we combine for our comparisons.
2. **S^3-NeRF** [46] is a recent method for learning neural scene representations using shadows. Using single-view RGB images captured under varying illumination, it trains a neural SDF model by exploiting shadow and shading information. A sphere is initialized at the origin where the object is assumed to be and known camera and light positions are used to model the scene's bidirectional reflectance distribution function. S^3-NeRF reconstructs both the object casting shadows and all other background scene geometry, making it a suitable comparison.

Metrics. We use L1 depth error to evaluate our method for 3D reconstruction, as done in past work [14, 19, 46]. In addition, we also report PSNR on reconstructed depth images. Since BF Lidar reconstructs a PC, we also include metrics on Chamfer distance. To convert the BF Lidar PC to depth for depth metrics, we increase the size of each point and project the depth to the test view, taking the smallest depth value along each ray. We find this produces better

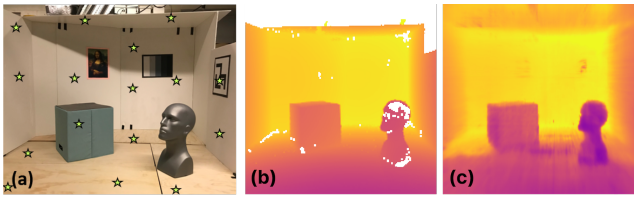


Figure 5. **Real-World Results. (a)** Captured scene (stars are illumination spots), **(b)** BF Lidar result, **(c)** PlatoNeRF result. Our method achieves similar results as BF Lidar, with much fewer artifacts.

results than first converting to a mesh before rendering depth.

Simulated Results. Depth metrics for both the training view and across 120 test views for both our method and the baseline methods are reported in Table 1, and Chamfer distance is reported in Table 2. We find that our method consistently outperforms both BF Lidar and S^3-NeRF across both sets of metrics. Qualitative results are shown in Figure 4. Our method is able to reconstruct the visible and occluded parts of the scene, providing accurate scale and absolute depth. Due to our use of an implicit representation, we achieve much smoother results than BF Lidar. However, because our method uses vanilla NeRF, there are small floaters visible in some results. In contrast, S^3-NeRF uses an SDF representation, reducing floaters, but resulting in overly smoothed results, e.g. the dragon's head.

Despite there being no two-bounce signal for any illumination point in the area directly behind the object, both PlatoNeRF and S^3-NeRF interpolate in this area. Since BF Lidar is not learning-based, the lack of two-bounce signal in this area results in a hole. We do not include this area in our depth metrics. We note that PlatoNeRF also accurately extrapolates beyond the camera's field of view, in parts of the scene close to the camera. Finally, since RGB based shape from shadow methods, such as S^3-NeRF, produce relative rather than absolute depth, we tried running iterative closest point [2] on S^3-NeRF's unprojected depth map, but found it does not improve S^3-NeRF's metrics. We also note that, as in the original work, we train S^3-NeRF with RGB images rendered with only one bounce, as we found it does not converge when trained on images rendered with multi-bounce.

Real-World Results. We show real-world results in Figure 5 and compare with BF Lidar. PlatoNeRF method achieves competitive performance. While BF Lidar produces many artifacts, especially near edges and specular areas on the mannequin, PlatoNeRF produces far fewer, despite not modeling specular surfaces. In general, PlatoNeRF produces smoother depth, but small floaters are noticeable, especially in the nearby floor region, which is an area for future work.

4.3. Ablations

We ablate our method to understand how it is affected by (1) reduced spatial and temporal resolution, (2) ambient light, (3) background albedo, (4) non-planar surfaces, and (5) number of illum. points. (1) is ablated in comparison to BF Lidar to highlight the benefits of our method over related lidar work. (2) and (3) are ablated in comparison to S^3-NeRF to highlight the fundamental advantages of using lidar compared to RGB when measuring shadows. Finally, (4) and (5) are done only on PlatoNeRF. All ablations are done on the chair scene.

Table 3. **Ablations on Lidar Sensor.** Lidars on consumer devices have lower spatial- and temporal-resolution than research-grade lidars. We ablate the impact of these sensor parameters on our method and BF lidar and find that our method is much more generalizable due to the interpolation of our implicit representation.

Spatial Resolution		
	L1 Depth (m)	
Downsample	Ours	BF Lidar
128×128	**0.0880**	0.1236
64×64	0.0932	0.1759
32×32	0.1070	0.1799

Temporal Resolution		
	L1 Depth (m)	
Upsample	Ours	BF Lidar
256 ps	**0.0965**	0.2802
512 ps	0.1210	0.3119
1024 ps	0.1833	0.3510

Table 4. **Ablations on Scene Properties.** We observe that RGB methods that exploit shadows are sensitive to scene properties that affect the visibility of the shadow, notably ambient light and background albedo. We ablate our method with S^3-NeRF as we vary these properties and note that, while S^3-NeRF is relatively robust, PlatoNeRF is much more so due to our use of a lidar sensor.

Ambient Light		
	L1 Depth (m)	
Intensity	Ours	S^3–NeRF
0	0.0862	0.1178
4	**0.0794**	0.3080

Scene Albedo		
	L1 Depth (m)	
Albedo	Ours	S^3–NeRF
$0\times$ less	0.0862	0.1178
$4\times$ less	**0.0859**	0.2152

4.3.1 Spatial and Temporal Resolution

In comparison to research-grade lidars, lidars on consumer devices have lower spatial and temporal resolution. We highlight an advantage of PlatoNeRF over BF Lidar, which is better generalization to low-resolution regimes due to our implicit representation. Quantitative and qualitative results are shown in Table 3 and Figure 6 (rows 1–2), respectively.

Spatial Resolution. To study spatial resolution, we downsample the number of pixels by four, eight, or sixteen, keeping field of view the same. Resulting spatial resolutions are 128×128, 64×64, and 32×32. We find that BF Lidar's accuracy degrades more significantly than PlatoNeRF since there is no mechanism to interpolate across missing pixels. Even at 32×32 resolution, PlatoNeRF accurately reconstructs the scene, albeit with slightly more artifacts. More in our supp.

Temporal Resolution. We increase the bin size of our transients from 128 ps to 256 ps, 512 ps, and 1024 ps by integrating the intensities within each bin, resulting in fewer bins per transient, and thus less precise depth information. We find that depth error for BF Lidar degrades more than

PlatoNeRF. Surfaces predicted by BF Lidar become rough and uneven due to ambiguity in the depth per pixel (see our supplement). In contrast, while PlatoNeRF's predicted depth becomes less accurate as bin size increases, it maintains smooth geometry and consistency between nearby pixels.

4.3.2 Ambient Light and Low Albedo Backgrounds

In real-world settings, there may be high ambient light or low scene albedo, both of which make detection of shadows in RGB images more challenging. In contrast, lidar-based methods, such as PlatoNeRF, are fundamentally more robust to these low signal-to-noise (SNR) and signal-to-background (SBR) scenarios. While ambient light and low albedo impact both RGB and lidar-based approaches, lidar-based approaches can mitigate their impact with *time- and wavelength-gating*. Ambient noise is uniformly distributed over time. Time gating enables suppression of ambient noise by only considering timing bins containing the pulse signal. Wavelength gating enables suppression of ambient noise in wavelengths not measured by the SPAD sensor, whereas RGB sensors have broadband sensitivity to visible wavelengths. A detailed explanation of the gating principle is provided in the supplement. Empirical results for these ablations are reported in Table 4.

Ambient Light. Increasing the ambient light would increase the background and noise in the SBR or SNR term. We empirically validate that PlatoNeRF is able to handle ambient light in the scene, while S^3-NeRF depth error increases. For this experiment, we render the scene with an added area light to train both methods. The area light intensity is the same in both RGB and lidar rendering. We do not model saturation or pileup distortion [30] effects, since these are not significant in indoor environments. PlatoNeRF's reconstruction under ambient light is shown in Figure 6 (row 3) and S^3-NeRF's is in the supplement.

Albedo. Reducing the albedo would reduce the signal in the SNR and SBR term. However, unlike RGB sensors, scene albedo has a minor effect on lidar. To show the impact of albedo on PlatoNeRF and S^3-NeRF, we reduce the albedo of all background surfaces. While the shadow is still discernible to the human eye (see supplement), the lowered albedo causes S^3-NeRF's depth error to increase, while PlatoNeRF is unaffected. S^3-NeRF produces accurate depth from the training view, but, in occluded regions, geometry is missing due to the weaker shadows.

4.3.3 Other Ablations

Lastly, we study the impact of both non-planar background geometry and number of illumination spots on PlatoNeRF reconstructions. The qualitative results for both are shown in Figure 6 (row 3). The non-planar scene consists of curved walls on either side. PlatoNeRF's depth L1 is 7.75 cm and PSNR is 27.25 dB across test views, similar to previous experiments. For our illumination spot ablation, we reduce the number of illumination spots from 16 to 8, leading to L1 error of 9.12 cm and PSNR of 26.33 dB across test views, a small drop from when all sixteen views are used. Varying illumination spots are studied more in the supplement.

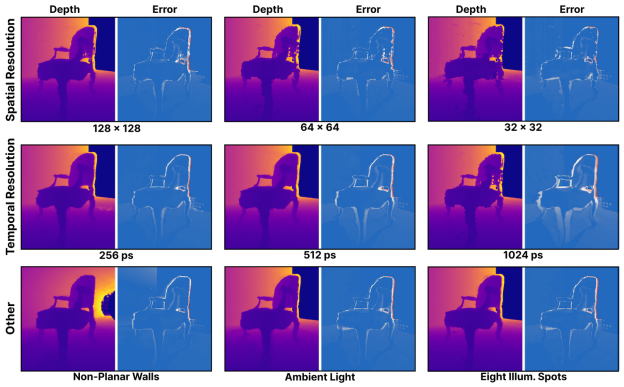


Figure 6. **Ablations.** We study the impact of spatial and temporal resolution on PlatoNeRF, finding that the scene is well reconstructed despite large degradation to both. While depth is visually similar for different temporal resolutions, the error maps indicate increasing displacement of the chair. The last row shows results for non-planar walls, ambient light, and fewer illumination points.

5. Conclusion

We present a method for reconstructing lidar measurements with NeRF, which enables physically-accurate 3D geometry to be learned from a single view. We illuminate the scene with a pulsed laser and record the two-bounce time of flight. This data is used to supervise NeRF, which is trained to learn the optical path of two-bounce light. Our method outperforms related work in single-view 3D reconstruction, reconstructs scenes with fully occluded objects, and learns metric depth from any view. Lastly, we demonstrate generalization to varying sensor parameters and scene properties.

Limitations. Our method has a couple limitations. First, we only model Lambertian reflectance. Second, our method is built on top of vanilla NeRF, and, as a result, occasionally has floaters. However, our method is agnostic to the flavor of NeRF and can be integrated into others in the future.

Future Work. We believe this research is a promising direction as lidar becomes ubiquitous. Future directions enabled by PlatoNeRF include incorporating RGB and lidar with neural rendering, incorporating data priors that are commonly used for single-view 3D reconstruction into PlatoNeRF, and modeling more than two-bounces of light.

Acknowledgements. We thank Akshat Dave for his paper feedback and insights into time of flight imaging and we thank Wenqi Yang for her guidance with S^3-NeRF.

References

[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. In *NeurIPS*, 2021. 3

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 7

[3] Clara Callenberg, Zheng Shi, Felix Heide, and Matthias B Hullin. Low-cost spad sensing for non-line-of-sight tracking, material classification and depth imaging. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 2

[4] Edoardo Charbon. Introduction to time-of-flight imaging. In *SENSORS, 2014 IEEE*, pages 610–613. IEEE, 2014. 4

[5] Yuki Fujimura, Takahiro Kushida, Takuya Funatomi, and Yasuhiro Mukaigawa. Nlos-neus: Non-line-of-sight neural implicit surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10532–10541, 2023. 3

[6] Connor Henley, Tomohiro Maeda, Tristan Swedish, and Ramesh Raskar. Imaging behind occluders using two-bounce light. In *ECCV*, 2020. 3

[7] Connor Henley, Joseph Hollmann, and Ramesh Raskar. Bounce-flash lidar. *IEEE Transactions on Computational Imaging*, 8:411–424, 2022. 2, 3, 6, 1

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[9] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. arXiv:2311.04400, 2023. 1, 2

[10] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural lidar fields for novel view synthesis. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[11] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[12] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 5, 1

[13] Suren Jayasuriya, Adithya Pediredla, Sriram Sivaramakrishnan, Alyosha Molnar, and Ashok Veeraraghavan. Depth fields: Extending light field techniques to time-of-flight imaging. In *2015 International Conference on 3D Vision*, pages 1–9. IEEE, 2015. 2

[14] Asaf Karnieli, Ohad Fried, and Yacov Hel-Or. Deepshadow: Neural shape from shadow. In *European Conference on Computer Vision*, pages 415–430. Springer, 2022. 2, 6

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. Looking around the corner using transient imaging. In *2009 IEEE 12th International Conference on Computer Vision*, pages 159–166. IEEE, 2009. 3

[17] José Luis Landabaso, Montse Pardàs, and Josep Ramon Casas. Shape from inconsistent silhouette. *Comput. Vis. Image Underst.*, 112:210–224, 2008. 2

[18] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[19] Jingwang Ling, Zhibo Wang, and Feng Xu. Shadowneus: Neural sdf reconstruction by shadow ray supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2023. 1, 2, 6

[20] Ruoshi Liu, Sachit Menon, Chengzhi Mao, Dennis Park, Simon Stent, and Carl Vondrick. Shadows shed light on 3D objects. arXiv:2206.08990, 2022. 1, 2

[21] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. arXiv:2303.11328, 2023. 1, 2

[22] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (Proceedings of SIGGRAPH)*, 21(4):163–169, 1987. 5

[23] Anagh Malik, Parsa Mirdehghan, Sotiris Nousias, Kiriakos N. Kutulakos, and David B. Lindell. Transient neural radiance fields for lidar view synthesis and 3D reconstruction. arXiv:2307.09555, 2023. 3

[24] Worthy N. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2): 150–158, 1983. 2

[25] Andreas Meuleman, Hakyeong Kim, James Tompkin, and Min H Kim. Floatingfusion: Depth from tof and image-stabilized stereo cameras. In *European Conference on Computer Vision*, pages 602–618. Springer, 2022. 3

[26] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 5

[27] Fangzhou Mu, Sicheng Mo, Jiayong Peng, Xiaochun Liu, Ji Hyun Nam, Siddeshwar Raghavan, Andreas Velten, and Yin Li. Physics to the rescue: Deep non-line-of-sight reconstruction for high-speed imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[29] Adithya Pediredla, Ashok Veeraraghavan, and Ioannis Gkioulekas. Ellipsoidal path connections for time-gated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 6

[30] Adithya K Pediredla, Aswin C Sankaranarayanan, Mauro Buttafava, Alberto Tosi, and Ashok Veeraraghavan. Signal processing based pile-up compensation for gated single-photon avalanche diodes. *arXiv preprint arXiv:1806.07437*, 2018. 8

[31] PicoQuant. LDH series picosecond pulsed diode laser heads, 2023. 1

[32] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2

[33] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. arXiv:2310.17994, 2023. 2

[34] Silvio Savarese, Holly Rushmeier, Fausto Bernardini, and Pietro Perona. Shadow carving. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 190–197. IEEE, 2001. 2, 6

[35] Siyuan Shen, Zi Wang, Ping Liu, Zhengqing Pan, Ruiqian Li, Tian Gao, Shiying Li, and Jingyi Yu. Non-line-of-sight imaging via neural transient fields. *TPAMI*, 43(7):2257–2268, 2021. 3

[36] Daniel Smith. Calculating the emission spectra from common light sources, 2016. 1, 2

[37] Siddharth Somasundaram, Akshat Dave, Connor Henley, Ashok Veeraraghavan, and Ramesh Raskar. Role of transients in two-bounce non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9192–9201, 2023. 3

[38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[39] Tang Tao, Longfei Gao, Guangrun Wang, Peng Chen, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. LiDAR-NeRF: Novel lidar view synthesis via neural radiance fields. *arXiv preprint arXiv:2304.10406*, 2023. 3

[40] Kushagra Tiwary, Tzofi Klinghoffer, and Ramesh Raskar. Towards learning neural representations from shadows. In *European Conference on Computer Vision*, pages 300–316. Springer, 2022. 1, 2

[41] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3(1): 745, 2012. 3

[42] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 2

[43] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[44] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. SinNeRF: Training neural radiance fields on complex scenes from a single image. In *ECCV*, 2022. 1

[45] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2

[46] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. S^3-NeRF: Neural reflectance field from shading and shadow under a single viewpoint. In *NeurIPS*, 2022. 1, 2, 6

[47] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[48] Junzhe Zhu and Peiye Zhuang. HiFA: High-fidelity text-to-3D with advanced diffusion guidance. arXiv:2305.18766, 2023. 1

PlatoNeRF: 3D Reconstruction in Plato's Cave via Single-View Two-Bounce Lidar

Supplementary Material

A. Time & Wavelength Gating in Lidar

As described in the main text, PlatoNeRF (and lidar-based methods) offer fundamental advantages over RGB-based methods in practical scenarios with uncontrolled scene albedos and ambient illumination. Lidars can leverage their picosecond timing resolution for *time gating* to enhance signal-to-background ratio (SBR) of measured shadow images. In addition, unlike RGB sensors, lidar sensors do not require wideband spectral sensitivity. Therefore, ambient illumination that has different wavelength than that of the laser's can be suppressed using *wavelength gating*.

The principle of time gating is illustrated in Figure 7. A measured lidar signal $i(t)$ can be decomposed into the pulse signal $s(t)$ and (roughly) constant ambient background noise $n(t) = N$. An RGB sensor would integrate over this timing information and measure

$$i = \int_0^T i(t)dt = \int_0^T s(t) + n(t)dt \tag{9}$$

$$= \int_0^T s(t)dt + NT, \tag{10}$$

where T is the length of the transient signal. The measurement i results in a SBR of

$$\text{SBR} = \frac{\int_0^T s^2(t)dt}{N^2 T} \tag{11}$$

On the other hand, a lidar sensor would only use relevant parts of the transient, i.e., around the signal peak. A time-gated lidar would therefore measure

$$i = \int_{T_1}^{T_2} s(t)dt + NT, \tag{12}$$

$$\text{with SBR}_{\text{gated}} = \frac{\int_{T_1}^{T_2} s^2(t)dt}{N^2 W}, \tag{13}$$

where T_1 and T_2 determine the gated window in the transient signal and $W = T_2 - T_1$ is the window size. Note that the numerator of Equation 11 is roughly the same as the SBR in Equation 13 because $s(t) \approx 0$ for $t < T_1$ and $t > T_2$, as shown in Figure 7(a). Therefore, time gating offers an SNR improvement of $\frac{T}{W}$ over techniques that leverage RGB or intensity signals. Note that the SBR enhancement is inversely proportional to the gated window. We do not account for Poisson noise effects, which, in practice, would introduce trade-offs in determining the window size. Empirical results are plotted in Figure 7(b)-(d) on the effects of time gating on enhancing contrast in shadow images.

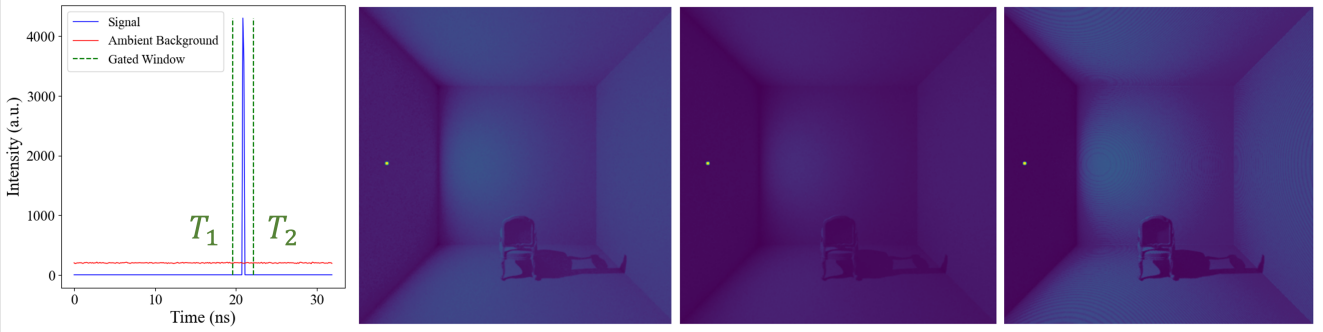
A similar idea can be applied to gate wavelengths. Most of the signal will be concentrated within a narrow spectral range, and all other intensities can be gated out with a narrow-band pass filter, as shown in Figure 8. This figure plots the emission spectra of an LED light [36] and the gating profile is determined by a 685 nm PicoQuant pulsed laser [31].

B. Simulated Dataset Details

In this section, we describe the simulated datasets that we render and use to compare our method to past work in more detail. We render four simulated scenes, as described in the main text, with both a lidar and RGB camera in Mitsuba [12]. The lidar data is used to run PlatoNeRF and Bounce Flash (BF) Lidar [7] and the RGB data is used to run S^3-NeRF [46]. The same sixteen scene points are illuminated in both the lidar and RGB data. In the lidar data, the sixteen points are illuminated with a laser and, in the RGB data, point light sources are placed at each of the sixteen points. A camera to world transform from OpenGL (x right, y up, z back) to Mitsuba (x left, y up, and z forward) is used to train each method with this data. Ground truth depth for both the train view and 120 test views are provided. A subset of the test view frames are shown in the video results on the project page. We plan to release all data for use in future work.

Lidar Data. The lidar (direct time of flight) data is rendered at 512×512 spatial resolution with a temporal resolution (bin size) of 128 ps. We simulate a laser by using a spot light source and setting the cutoff angle as 0.2 and the beam width as 0.1. To choose the illumination points, we randomly illuminate twenty four points in the scene and then heuristically choose sixteen that maximize diversity.

RGB Data. To compare with S^3-NeRF, we render each scene with both lidar (to run our method) and RGB (to run S^3-NeRF) in Mitsuba. When rendering with RGB, we compute the location of the scene point where the laser first hits the scene and place a point light source at this location. By placing point light sources at the same location as where the laser hits the scene, we ensure the same shadows are cast in the scene in both the lidar and RGB data. RGB images are rendered with max depth to set to 2, ensuring only first-bounce light is rendered, as required by S^3-NeRF. Rendered images are gamma corrected prior to training.



(a) Histogram (b) Noiseless (c) Ambient Noise (d) Ambient Noise (Time Gated)

Figure 7. **Time Gating with Lidar.** (a) A transient is plotted at a single pixel. Note that most of the signal (blue) is concentrated within a few timing bins $\sim$ 20 ns. By only gating a window (green) around the signal, most of the noise profile (red) can be suppressed. (b)-(d) Measured intensity images without time gating (b, c) and with time gating (d).

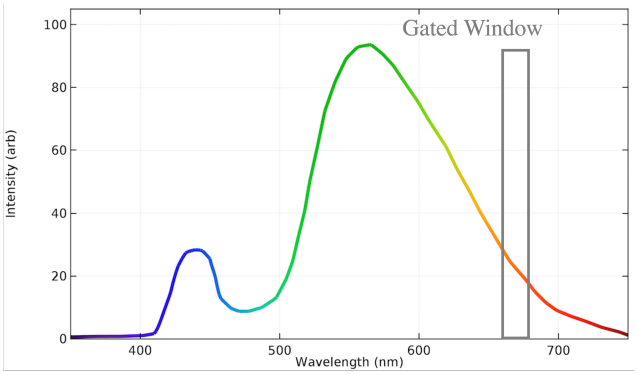


Figure 8. **Wavelength Gating.** Ambient illumination under an LED light is compared to the spectral gating window needed for a spectral window centered at 685 nm. Figure adapted from [36].

C. Training Details

Reproducibility. We will release all data, code, and model checkpoints, along with documentation, prior to camera ready to ensure our work is fully reproducible by others. Code is provided in the "code" folder of the supplement. Simulated data is rendered in Mitsuba world coordinates and PlatoNeRF uses OpenGL camera coordinates.

PlatoNeRF. We train our model for 200k iterations. For the first 25k iterations, only the distance loss is applied, while both the distance and shadow losses are applied thereafter. We use a threshold of 15% on the shadow confidence map (computed as the maximum of the cross-correlation described in Section 3.3) when extracting ground truth shadow masks from the raw lidar measurements. This threshold is used across all experiments, except the ambient light experiment, where we further tune it.

Bounce Flash Lidar. Bounce Flash (BF) Lidar consists of two steps: (1) estimating visible geometry via constraints on ellipsoidal geometry, and (2) estimating occluded geometry with shadow carving. For each scene, we run a grid search over thresholds for shadow extraction and occupancy probability (applied to the occupancy probabilities predicted from shadow carving) to maximize BF Lidar accuracy.

S^3-NeRF. We found the default training parameters provided for S^3-NeRF work the best on our data. We only modify the light intensity parameter to match our rendered data when training. When training with ambient light, we run a grid search over the ambient light intensity (amb_i) parameter to maximize S^3-NeRF reconstruction quality, but find that under a reasonably high ambient area light, S^3-NeRF is not able to reconstruct the scene regardless of this parameter.

Table 5. **Ablations on Number of Illumination Points.** We study how varying the number of illumination points between two and sixteen impacts PlatoNeRF reconstruction quality.

Illumination Spots		
	PlatoNeRF	
# Spots	L1 (m)	PSNR (dB)
16	**0.0862**	**26.58**
8	0.0912	26.33
4	0.1347	25.15
2	0.2147	21.61

D. Extended Ablations

In this section, we add further detail and discussion on the results of our ablation, quantitatively reported in the main text. In addition, we provide further ablation on the impact of the number of illumination points on PlatoNeRF (Table 5).

D.1. Spatial- and Temporal-Resolution

Qualitative results comparing PlatoNeRF and Bounce Flash (BF) Lidar under varying spatial- and temporal-resolutions are shown in Figure 9. This ablation is important because

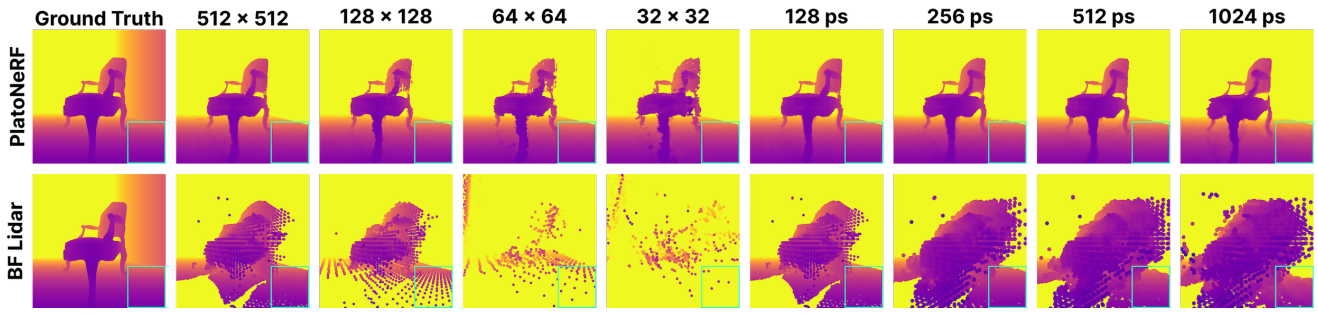


Figure 9. **Spatial- and Temporal-Resolution Ablation.** We compare PlatoNeRF and Bounce Flash (BF) Lidar as spatial- and temporal-resolution is reduced. PlatoNeRF continues to produce smooth geometry in both cases, whereas BF Lidar produces sparse geometry when spatial resolution is reduced and bumpy geometry when temporal resolution is reduced, as highlighted in the area in the green boxes.

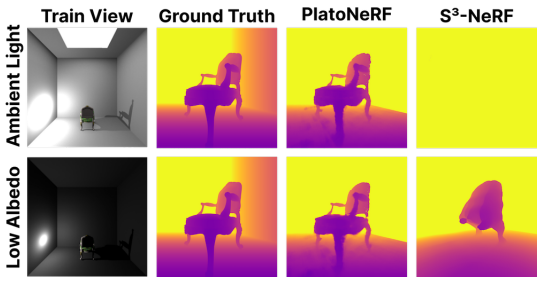


Figure 10. **Ambient Light and Low Albedo Background Ablation.** We compare PlatoNeRF and S^3-NeRF when trained on a scene with either ambient light or a low albedo background. PlatoNeRF is robust to both, whereas the performance of S^3-NeRF degrades.

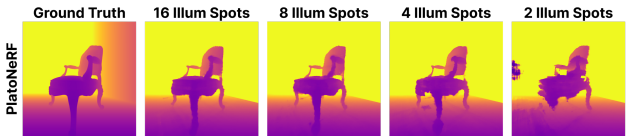


Figure 11. **Illumination Point Ablation.** We ablate the impact of varying the number of illumination points between two and sixteen on PlatoNeRF. While more illumination points improves reconstruction quality, the chair's geometry is still coarsely reconstructed with just two illumination points.

lidars on consumer devices are often constrained to much lower resolutions than research-grade lidars. Spatial resolution is varied by downsampling the number of pixels, while keeping the field of view of the lidar the same. As spatial resolution is reduced, geometry predicted by BF Lidar becomes sparser. The depth estimation of visible points in the scene remains accurate, but there is no interpolation between these points. The sparsity in visible depth information negatively impacts the shadow carving step of BF Lidar, leading to poor reconstruction of the chair in lower spatial resolution regimes. On the other hand, because PlatoNeRF is able to smoothly interpolate across missing pixels, the resulting reconstruction is significantly more accurate.

Temporal resolution is related to the bin size of the transient (i.e. the amount of time between each lidar measurement). To increase the bin size and thus reduce the temporal resolution of the lidar, we integrate intensities within the bins. For example, when increasing bin size from 128 to 256 ps, we sum intensities for over every two bins. BF Lidar results maintain the shape of the chair (since shadow carving is not significantly affected), but the visible geometry becomes rough and bumpy since the supervision for the depth of each visible pixel is less precise. On the other hand, PlatoNeRF maintains smooth reconstructions.

D.2. Ambient Light

Qualitative results comparing PlatoNeRF and S^3-NeRF reconstructions under ambient light are shown in Figure 10 (top row). While S^3-NeRF is able to model small amounts of ambient light, it fails under realistic amounts of ambient light, in this case, from an added area light. On the other hand, PlatoNeRF is still able to accurately reconstruct the scene with the same ambient light added.

D.3. Low-Albedo Backgrounds

Qualitative results comparing PlatoNeRF and S^3-NeRF reconstructions with a low albedo background are shown in Figure 10 (bottom row). S^3-NeRF is able to accurately reconstruct the visible portion of the scene, but is unable to recover occluded geometry due to worse contrast in the shadow (though it is still discernible to the human eye, as shown in Figure 10). On the other hand, PlatoNeRF is not significantly affected by scene albedo due to its use of a lidar rather than RGB sensor.

D.4. Number of Illumination Points

We further ablate the impact of reducing the number of illumination points used to train PlatoNeRF. In our main experiments, we use sixteen illumination points. We reduce that number to eight, four, and two and report the results

in Table 5. Qualitative results are shown in Figure 11. The scene is reconstructed for each number of illumination points, however, as the number is reduced, quality also decreases, as there is less information about occluded regions. When there are only two illumination points, the occluded chair legs are not reconstructed. We note that while we study the number of illumination points, their location is also an important factor in reconstruction quality. As the number of illumination points is reduced, the location of the remaining illumination points becomes increasingly important, i.e. casting shadows with the most relevance and diversity. In these experiments, we randomly choose which illumination points to use.