# Supplementary Information:
# Steady-State Non-Line-of-Sight Imaging

Wenzheng Chen[1,2*] Simon Daneau[1,3] Fahim Mannan[1] Felix Heide[1,4]

[1]Algolux [2]University of Toronto [3]Université de Montréal [4]Princeton University

## 1. Setup and Prototype Details

In this section, we provide a detailed description of the setup components and calibration procedures.

### 1.1. Setup Details

We design the proposed illumination setup with the goal of illuminating a visible wall surface with a high-power collinear white light beam that can be steered to points on this wall using a dual-axis galvo system. This objective can be reached by several methods, but many of them are expensive or request a lot of laboratory time to get a decent beam shape. While white LED sources are inexpensive and can provide high illumination powers, they are area sources and hence the output beam is so wide that only very complicated and costly inverse beam expander setups achieve a collinear beam enough powerful. Alternatively, white light sources such as supercontinuum lasers or Quartz-Tungsten-Halogen illuminators (and other lamp types) coupled with a fiber optic light guides would allow to generate a collimated white beam, but only at a limited output power with existing hardware. Another problem using white sources results from the Gaussian beam limitation which restraints the minimum achievable beam size. Forcing a reduced beam diameter can result in sufficient performance on tiny propagation distances because the diffraction angle is different for each wavelength, causing chromatic distortion over long light paths. Our setup has a light propagation length bigger than 2 m, which means we need long spatial coherence provided by sources like lasers.

To tackle these competing design goals, we propose to recombine three inexpensive lasers with their central wavelength corresponding to the center of the camera color filter spectra. One red laser (635 nm - 300 mW), green laser (532 nm - 200 mW) and blue laser (450 nm - 200 mW) are collinearized by two cheap Thorlabs soda-lime hot/cold dichroic mirrors (FM02R and FM04R) placed with an incidence angle of 45°. These mirrors have a high reflection/transmission percentage (87% to 98%) that allows to maintains enough power in the generated white light beam. We white balance the source by adjusting the independent modulation of each laser power instead of white-balancing post-capture which leads to unfavorable

*The majority of this work was done while interning at Algolux.

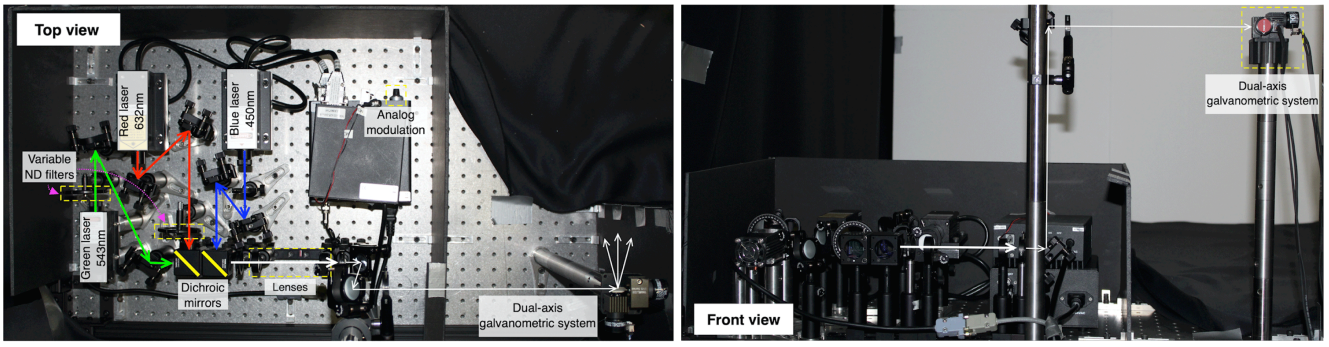


Figure 1. Prototype RGB laser source showing the path of the light, that ends in a dual-axis galvo. Left: variable neutral density (ND) filters and one analog modulation is used to control the individual power of lasers. Long focal lenses combination and several irises reduce the beam diameter. Right: only the white light path is shown. The camera is placed right next to the galvo.

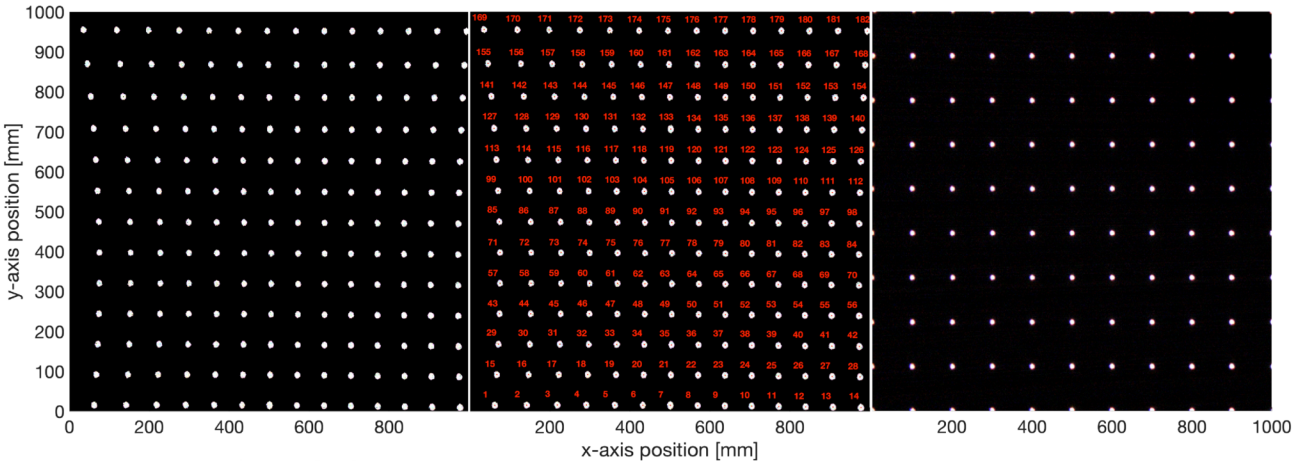


Figure 2. Dual-axis galvanometric system calibration procedure. From left to right: non-calibrated system driven by voltage intervals on both axes, the extracted and labeled laser points and uniformly sampled points after applying our calibration procedure.

noise statistics for unbalanced sources. To this end, we used for the green and red lasers, two Thorlabs round continuously variable neutral density (ND) filters (NDC-50C-2M) that have an optical density ranges between 0 and 2.0, which attenuates the transmitted beam to 1% of the incident intensity. The blue laser power is fine-tuned by an analog modulator. Also, the white balance is adjusted with a diffusing filter fixed after the galvo to spread the beam on the screen. The beam size just after the two dichroic mirror is too large to be directly fed into our dual-axis galvanometric system which works with a beam that have less than 5 mm diameter. We reduce the beam-size, at the cost of also reducing the power, by introducing irises and a pair of collimating lenses before the galvo. Note also that our Thorlabs GVS102 dual-axis galvo system with angular resolution of 15 $\mu$rad is more accurate than the cm resolution needed in practice.

### 1.2. Calibration Procedure

We calibrate the camera intrinsic and extrinsic parameters using 20 captures of a checkerboard calibration target and the popular Matlab Camera Calibrator toolbox. To calibrate our galvo, we sample the input voltage for both mirrors uniformly, as shown in Fig. 2. The pillow-shaped distortion is a result of the setup geometry and the distance between the two mirrors, where a given light path hits both mirrors sequentially, first getting oriented in vertical and then in the horizontal output direction. Hence, the x-axis position is depending on both mirror orientation. We compute the centroid of an average capture of all uniform-voltage laser points, illustrated in the middle plot of Fig. 2. Every spot is then associated with their corresponding x-axis and y-axis voltages, resulting in sparsely sampled lookup functions for the x and y direction of the galvo. Finally, we fit a third-degree polynomial to both functions which allows us to retrieve galvo voltages a given spatial target position on the diffuse wall. We measure an accuracy of 5 mm on the $1m \times 1m$ diffuse wall area.

## 2. Derivations for Planar Scene Recovery

### 2.1. Reflection Point

In this section, we derive the analytic gradients for the planar geometry estimation method from indirect reflections described in Sec. 4 of the main draft. To this end, we first derive Eq (4) from the main draft, which defines a point $\mathbf{p}$ that produces an indirect specular lobe around $\mathbf{c}$ for a given virtual source position $\mathrm{l}$ as

$$\mathbf{p}(\mathbf{v},\mathbf{n}) = \mathbf{c} + ((\mathbf{v}-\mathbf{c})\cdot\mathbf{n})\left(\mathbf{n} - \frac{\mathbf{v}-\mathrm{l}-((\mathbf{c}-\mathrm{l})\cdot\mathbf{n})\mathbf{n}}{\mathbf{n}\cdot(2\mathbf{v}-\mathbf{c}-\mathrm{l})}\right), \tag{1}$$

which is given as a function of the plane normal $\mathbf{n}$ and plane position $\mathbf{v}$. Note that the operator $\cdot$ represents the inner (dot-product) here. To derive this result, we consider the right triangles $\overline{\mathrm{lpl}_\perp}$ and $\overline{\mathbf{cpc}_\perp}$, where $\mathrm{l}_\perp$ and $\mathbf{c}_\perp$ are the orthogonal projections of $\mathrm{l}$ and $\mathbf{c}$ onto the plane, respectively. As the angle between the plane normal and the vectors $\mathrm{l}-\mathbf{p}$ and $\mathbf{c}-\mathbf{p}$

are equal, these two triangles are similar. Hence, the distance between $\mathbf{l}_\perp$ and $\mathbf{c}_\perp$ on the plane can be expressed as

$$\begin{aligned} \|\mathbf{l}_\perp - \mathbf{c}_\perp\| &= \|\mathbf{c} - \mathbf{l} + \mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\| = k + k\frac{(\mathbf{p}-\mathbf{l})\cdot\mathbf{n}}{(\mathbf{c}-\mathbf{l})\cdot\mathbf{n}} \quad \text{with} \quad k \in \mathbb{R}_+ \\ k &= \frac{\|\mathbf{c} - \mathbf{l} + \mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\|}{1 + \frac{(\mathbf{p}-\mathbf{l})\cdot\mathbf{n}}{(\mathbf{c}-\mathbf{l})\cdot\mathbf{n}}} \end{aligned} \tag{2}$$

where the positive scalar variable $k$ is the scale between both triangles. With this scale known, we can express the point $\mathbf{p}$, starting from $\mathbf{c}$, as

$$\begin{aligned} \mathbf{p}(\mathbf{v},\mathbf{n}) &= \mathbf{c} + ((\mathbf{v}-\mathbf{c})\cdot\mathbf{n})\mathbf{n} - \frac{\mathbf{c}-\mathbf{l}+\mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})}{\|\mathbf{c}-\mathbf{l}+\mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\|}\,\frac{\|\mathbf{c}-\mathbf{l}+\mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\|}{1+\frac{(\mathbf{p}-\mathbf{l})\cdot\mathbf{n}}{(\mathbf{c}-\mathbf{l})\cdot\mathbf{n}}} \\ &= \mathbf{c} + ((\mathbf{v}-\mathbf{c})\cdot\mathbf{n})\mathbf{n} - \frac{(\mathbf{c}-\mathbf{l}+\mathbf{n}((\mathbf{c}-\mathbf{l})\cdot\mathbf{n}))\,((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})}{(\mathbf{c}-\mathbf{l})\cdot\mathbf{n}+(\mathbf{p}-\mathbf{l})\cdot\mathbf{n}}, \end{aligned} \tag{3}$$

which immediately yields Eq. (1).

## 2.2. Analytic Gradient

The Jacobian of $\mathbf{p}$ from Eq. (1) has the following analytic form

$$\begin{aligned} \frac{\partial\mathbf{p}(\mathbf{v},\mathbf{n})}{\partial\mathbf{n}} =& (\mathbf{v}-\mathbf{c})\left(\mathbf{n} - \frac{\mathbf{v}-\mathbf{l}-((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\mathbf{n}}{\mathbf{n}\cdot(2\mathbf{v}-\mathbf{c}-\mathbf{l})}\right)^T + \\ & ((\mathbf{v}-\mathbf{c})\cdot\mathbf{n})\left(\mathbb{I} - \frac{\left(-((\mathbf{v}-\mathbf{l})\mathbf{n}^T) - \mathbf{n}^T(\mathbf{v}-\mathbf{l})\mathbb{I}\right)\mathbf{n}\cdot(2\mathbf{v}-\mathbf{c}-\mathbf{l}) - (\mathbf{v}-\mathbf{l}-((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\mathbf{n})\,(2\mathbf{v}-\mathbf{c}-\mathbf{l})^T}{(\mathbf{n}^T(2\mathbf{v}-\mathbf{c}-\mathbf{l}))^2}\right) \end{aligned} \tag{4}$$

$$\frac{\partial\mathbf{p}(\mathbf{v},\mathbf{n})}{\partial\mathbf{v}} = \mathbf{n}\left(\mathbf{n} - \frac{\mathbf{v}-\mathbf{l}-((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\mathbf{n}}{\mathbf{n}\cdot(2\mathbf{v}-\mathbf{c}-\mathbf{l})}\right)^T + ((\mathbf{v}-\mathbf{c})\cdot\mathbf{n})\left(\frac{2\mathbf{n}\,(\mathbf{v}-\mathbf{l}-((\mathbf{c}-\mathbf{l})\cdot\mathbf{n})\mathbf{n})^T}{(\mathbf{n}^T(2\mathbf{v}-\mathbf{c}-\mathbf{l}))^2}\right), \tag{5}$$

With these Jacobians in hand, we can derive the gradient of the objective function in the minimization problem from Eq. (6) from the main draft, which we list again for convienience

$$\begin{aligned} \theta^*,\phi^*,\nu^* &= \underset{\theta,\phi,\nu}{\operatorname{argmin}} \underbrace{\sum_{f=1}^{\tilde{F}}\sum_{i\in\Psi_f}\left\|\mathbf{p}_i^f(\mathbf{v},\mathbf{n}) - \overline{\mathbf{p}^f}\right\|_2^2}_{\Lambda(\theta,\phi,\nu)} \\ &\text{with} \quad \overline{\mathbf{p}^f} = \frac{1}{|\Psi_f|}\sum_{i\in\Psi_f}\mathbf{p}_i^f(\mathbf{v},\mathbf{n}), \end{aligned} \tag{6}$$

where we introduce an objective function shortcut $\Lambda$ for the objective itself. Recalling the normal and plane position parametrization

$$\mathbf{n}(\theta,\phi) = \begin{bmatrix}\cos(\theta)\,\sin(\phi)\\ \sin(\theta)\,\sin(\phi)\\ \cos(\phi)\end{bmatrix}, \quad \mathbf{v}(\nu) = \vec{o} + \nu\begin{bmatrix}0\\0\\1\end{bmatrix}, \tag{7}$$

we derive the partials of $\Lambda$ using the chain-rule as follows

$$\frac{\partial\Lambda(\theta,\phi,\nu)}{\partial\theta} = 2\sum_{f=1}^{\tilde{F}}\sum_{i\in\Psi_f}\left(\mathbf{p}_i^f(\mathbf{v}(\nu),\mathbf{n}(\theta,\phi)) - \overline{\mathbf{p}^f}\right)^T\left(\frac{\partial\mathbf{p}_i^f(\mathbf{v},\mathbf{n})}{\partial\mathbf{n}} - \frac{\partial\overline{\mathbf{p}^f}}{\partial\mathbf{n}}\right)\frac{\partial\mathbf{n}}{\partial\theta} \tag{8}$$

$$\frac{\partial\Lambda(\theta,\phi,\nu)}{\partial\phi} = 2\sum_{f=1}^{\tilde{F}}\sum_{i\in\Psi_f}\left(\mathbf{p}_i^f(\mathbf{v}(\nu),\mathbf{n}(\theta,\phi)) - \overline{\mathbf{p}^f}\right)^T\left(\frac{\partial\mathbf{p}_i^f(\mathbf{v},\mathbf{n})}{\partial\mathbf{n}} - \frac{\partial\overline{\mathbf{p}^f}}{\partial\mathbf{n}}\right)\frac{\partial\mathbf{n}}{\partial\phi} \tag{9}$$

$$\frac{\partial\Lambda(\theta,\phi,\nu)}{\partial\nu} = 2\sum_{f=1}^{\tilde{F}}\sum_{i\in\Psi_f}\left(\mathbf{p}_i^f(\mathbf{v}(\nu),\mathbf{n}(\theta,\phi)) - \overline{\mathbf{p}^f}\right)^T\left(\frac{\partial\mathbf{p}_i^f(\mathbf{v},\mathbf{n})}{\partial\mathbf{v}} - \frac{\partial\overline{\mathbf{p}^f}}{\partial\mathbf{v}}\right)\frac{\partial\mathbf{v}}{\partial\nu}. \tag{10}$$

where the the partials for the average $\overline{\mathbf{p}^f}$ follow immediately from it's definition Eq. (6), given Eq. (5) and Eq. (4). The parameter gradients of the plane parametrization from Eq. (7) are

$$\frac{\partial \mathbf{n}(\theta,\phi)}{\partial \theta} = \begin{bmatrix} -\sin(\theta)\,\sin(\phi) \\ \cos(\theta)\,\sin(\phi) \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{n}(\theta,\phi)}{\partial \phi} = \begin{bmatrix} \cos(\theta)\,\cos(\phi) \\ \sin(\theta)\,\cos(\phi) \\ -\sin(\phi) \end{bmatrix}, \quad \frac{\partial \mathbf{v}(\nu)}{\partial \nu} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

The partials for $\Lambda$ from Eq. (8) - (10) can be efficiently computed in vectorized form (before reduction in the sum), as the Jacobian matrices are small $3 \times 3$ matrices.

## 3. Specular ADMM Algorithm

In this section, we derive the Alternating Method of Multipliers Method (ADMM) [2] to solve Eq. (7) in the main document which recovers specular reflectance from a set of $N$ observations $\mathbf{b_i}$ for $i \in \{1, \dots N\}$ corresponding to virtual source positions $\mathbf{l}_i$, given the estimated homography mappings $\mathbf{H}_i$ as described in the main text. To make this document self-contained, we list Eq. (7) again

$$\begin{aligned} \underset{\mathbf{x}}{\operatorname{argmin}} \;\; & -\sum_{i=1}^{N} \log\left( \, p(\mathbf{b_i}|\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}\mathbf{x}) \, \right) \; + \; \Gamma_{\text{TV}}\left(\mathbf{x}\right) \\ & \text{subject to} \quad 0 \leq \mathbf{x} \end{aligned} \tag{12}$$

which is a maximum a posteriori (MAP) estimation problem with a linear penalty and a Poissonian-Gaussian likelihood term

$$\begin{aligned} p\left(\mathbf{b}_i|\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}\mathbf{x}\right) = & \prod_j \frac{\left(\frac{\kappa}{E}\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}\mathbf{x}\right)^{\mathbf{b}_i^j} e^{-\left(\frac{\kappa}{E}\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}\mathbf{x}\right)^j}}{\mathbf{b}_i^j!} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left((\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}\mathbf{x})^j - \mathbf{b}_i^j\right)^2}{2\sigma^2}} \\ & \text{subject to} \quad 0 \leq \mathbf{x}. \end{aligned} \tag{13}$$

This joint Poissonian-Gaussian likelihood consists of a Poissonian and Gaussian component calibrated according to Eq. (3) from the main manuscript with Poissonian parameter $\kappa$ and Gaussian parameter $\sigma$. Both are assumed to be independent of each other and independent across all measurement pixels $j$ which are indicated as superscripts. The linear forward operator $\mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i}$ consists of the spatially varying convolution matrix $\mathbf{K}$ which blurs the specular reflectance based on the angular falloff $\beta$ and distance to the wall (encoded by the homography), and a subsequent warping matrix $\mathbf{W}$ which warps plane coordinates to the image plane and resamples the blurred specular reflectance coefficients using bi-linear interpolation. We assume a Gaussian specular lobe with standard deviation of $5°$ and precompute the convolution kernels for $\mathbf{K}^{\beta}_{\mathbf{H}_i}$ with a limited support of 100 pixels. The warping matrix and convolution matrix are represented implicitly and used as matrix-free functions in the solver implementation. For ease of notation, we denote

$$\mathbf{A}_i := \mathbf{W}_{\mathbf{H}_i}\mathbf{K}^{\beta}_{\mathbf{H}_i} \quad \forall i \in \{1, \dots N\}. \tag{14}$$

With these definitions, we follow [6] and define the negative log-likelihood of Eq. (13) as

$$\begin{aligned} -\log\left(p(\mathbf{b}_i|\mathbf{A}_i\mathbf{x})\right) = & \sum_j \chi\left(\left(\frac{\kappa}{E}\mathbf{A}_i\mathbf{x}\right)^j, \mathbf{b}_i^j\right) \; + \; \frac{1}{2\sigma^2}\left((\mathbf{A}_i\mathbf{x})^j - \mathbf{b}_i^j\right)^2 \\ & \text{with} \quad \chi(a,b) = a - b\log(a) + \mathcal{I}_{\mathbb{R}^+}(a), \end{aligned} \tag{15}$$

where $\mathcal{I}_{\mathbb{R}^+}(a)$ is the indicator function for the positive orthant and the point-wise log-likelihood function $\chi$ is derived in [6]. Next, we follow [6] and reformulate Eq. (12) as the following objective

$$\begin{aligned} \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \sum_i^N \underbrace{-\log\left(p\left(\mathbf{b}_i|\mathbf{z}_i\right)\right)}_{g_1(\mathbf{z}_i)} + \underbrace{\mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_{N+1}\right)}_{g_2(\mathbf{z}_{N+1})} + \underbrace{\lambda\|\mathbf{z}_{N+2}\|_1 + \lambda\|\mathbf{z}_{N+3}\|_1}_{g_3(\mathbf{z}_{N+2},\mathbf{z}_{N+3})} \\ \text{subject to} \quad & \underbrace{\begin{bmatrix} \mathbf{A}_i \\ \mathbf{I} \\ \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix}}_{\mathbf{K}} \mathbf{x} - \underbrace{\begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_{N+1} \\ \mathbf{z}_{N+2} \\ \mathbf{z}_{N+3} \end{bmatrix}}_{\mathbf{z}} = 0 \quad \forall i \in \{1, \dots N\} \end{aligned} \tag{16}$$

where the auxiliary slack variables $\mathbf{z}$ allow to split the optimization over the coupled objective components $g$ into a sequence of optimization problems that can be solved efficiently. The matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ are horizontal and vertical first-order derivative matrices, respectively. Moreover, we resolve the total variation (TV) prior [3] as an anisotropic TV prior with weight $\lambda$. To solve the reformulated optimization problem from Eq. (16), we use the linearized variant of ADMM [2] and solve the spectral reflectance estimation problem as where the parameter $\tau$ is a scalar algorithm parameter that controls the

**Algorithm 1** ADMM Algorithm for Planar Reflectance Estimation

1: **for** $k = 1$ **to** $M$
2: $\mathbf{x} = \operatorname{argmin} \ \frac{1}{2} \left\| \mathbf{K}\mathbf{x} - \mathbf{z} + \mathbf{u} \right\|_2^2$
3: $\mathbf{z}_i = \operatorname{argmin} \ -\log\left(p\left(\mathbf{b}_i|\mathbf{z}_i\right)\right) + \frac{\tau}{2} \left\| \mathbf{A}_i\mathbf{x} + \mathbf{u}_i - \mathbf{z}_i \right\|_2^2 \quad \forall i \in \{1, \dots N\}$
4: $\mathbf{z}_{N+1} = \operatorname{argmin} \ \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_{N+1}\right) + \frac{\tau}{2} \left\| \mathbf{x} + \mathbf{u}_{N+1} - \mathbf{z}_{N+1} \right\|_2^2$
5: $\mathbf{z}_{N+2} = \operatorname{argmin} \ \lambda \|\mathbf{z}_{N+2}\|_1 + \frac{\tau}{2} \left\| \mathbf{x} + \mathbf{u}_{N+2} - \mathbf{z}_{N+2} \right\|_2^2$
6: $\mathbf{z}_{N+3} = \operatorname{argmin} \ \lambda \|\mathbf{z}_{N+3}\|_1 + \frac{\tau}{2} \left\| \mathbf{x} + \mathbf{u}_{N+3} - \mathbf{z}_{N+3} \right\|_2^2$
7: $\mathbf{u} = \mathbf{u} + \mathbf{K}\mathbf{x} - \mathbf{z}$
8: **end for** ,

algorithm stepsize. We set all algorithm hyper-parameters adopting the settins from [7]. The ADMM Alg. 1 minimizes the Lagrangian of Eq. (16) in a coordinate descent approach, solving the objective for one of the (auxiliary) variables a the time, while keeping all other variables fixed. We refer the reader to [12, 2] for additional details. In the following, we derive the individual algorithm substeps for the optimization problems from line 2 to line 6.

### 3.1. Quadratic Subproblem in Algorithm 1

We solve the quadratic subproblem in line 2 of Alg. 1 using warm-started truncated conjugate gradient (CG). In contrast to recent methods solving for albedo volumes [10, 8, 11], we recover a compact 2D slice on the previously estimated planar object surface. We choose a resolution of $500 \ \times \ 500$ for the unknown $\vec{x}$ instead of the cubic resolutions required in for these existing methods. This allows for an efficient matrix-free solves of this suproblem using three iterations of the CG implementation from [7] with warm-start of $\mathbf{x}$ from the previous ADMM iteration. We have implemented the re-sampling and spatially-varying convolutions of the operator $\mathbf{A}$ and $\mathbf{A}^T$ using C++ code.

### 3.2. Poisson-Gaussian Subproblem in Algorithm 1

Introducing the free variable $\mathbf{v}$ for notational brevity, we follow [6] and formulate the subproblem in line 3 of Alg. 1 as

$$\operatorname*{argmin}_{\mathbf{z}_i} \ g_1(\mathbf{z}_i) + \frac{\tau}{2} \left\| \mathbf{v} - \mathbf{z}_1 \right\|_2^2, \tag{17}$$

which has the form of a proximal operator [12, 2] similar to the Poisson operator from [5]. Specifically, we derive

$$\begin{aligned}
&\operatorname*{argmin}_{\mathbf{z}_i} \ g_1(\mathbf{z}_i) + \frac{\tau}{2}\|\mathbf{z}_i - \mathbf{v}\|_2^2 \\
=&\operatorname*{argmin}_{\mathbf{z}_i \in \mathbb{R}^+} \ \underbrace{\mathbf{z}_i - \mathbf{b}_i \log(\mathbf{z}_i) + \frac{\tau}{2}\|\mathbf{z}_i - \mathbf{v}\|_2^2 + \frac{1}{2\sigma^2}\|\mathbf{z}_i - \mathbf{b}_i\|_2^2}_{\Upsilon(\mathbf{z}_i)} \\
\Leftrightarrow \ &\frac{\partial \Upsilon(\mathbf{z}_{i\text{opt}})}{\partial \mathbf{z}_i} = \mathbf{1} - \frac{\mathbf{b}_i}{\mathbf{z}_{i\text{opt}}} + \tau\mathbf{z}_{i\text{opt}} - \tau\mathbf{v} + \frac{1}{\sigma^2}\mathbf{z}_{i\text{opt}} - \frac{1}{\sigma^2}\mathbf{b}_i \stackrel{!}{=} 0 \quad \text{s.t.} \quad \mathbf{z}_{i\text{opt}} \in \mathbb{R}^+ \\
\Leftrightarrow \ &\mathbf{z}_{i\text{opt}}^2 + \left(\frac{\sigma^2}{\tau\sigma^2+1} - \frac{\tau\sigma^2}{\tau\sigma^2+1}\mathbf{v} - \frac{1}{\tau\sigma^2+1}\mathbf{b}_i\right) \cdot \mathbf{z}_{i\text{opt}} - \frac{\sigma^2}{\tau\sigma^2+1}\mathbf{b} = 0 \quad \text{s.t.} \quad \mathbf{z}_{i\text{opt}} \in \mathbb{R}^+ \\
\Leftrightarrow \ &\mathbf{z}_{i\text{opt}} = \frac{\left(\frac{\tau\sigma^2}{\tau\sigma^2+1}\mathbf{v} + \frac{1}{\tau\sigma^2+1}\mathbf{b}_i - \frac{\sigma^2}{\tau\sigma^2+1}\right)}{2} + \sqrt{\frac{\sigma^2}{\tau\sigma^2+1}\mathbf{b}_i + \frac{\left(\frac{\sigma^2}{\tau\sigma^2+1} - \frac{\tau\sigma^2}{\tau\sigma^2+1}\mathbf{v} - \frac{1}{\tau\sigma^2+1}\mathbf{b}_i\right)^2}{4}},
\end{aligned}$$

which is a root-finding problem of a second-order polynomial. Due to the positivity constraint the minimum is uniquely defined [5]. Finally, replacing the free variable $\mathbf{v}$ with the corresponding value for $\mathbf{A}_i\mathbf{x} + \mathbf{u}_i$ in line 4 of Alg. 1 yields the concrete algorithm step.

### 3.3. Indicator Function Subproblem in Algorithm 1

The subproblem in line 4 of Alg. 1 also has the form of a proximal operator [12, 2]. Following [12] this algorithm step is the projection operator defined for the free substitution variable $\mathbf{v}$ as

$$\underset{\mathbf{z}_{N+1}}{\operatorname{argmin}} \; \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_{N+1}\right) + \frac{\tau}{2}\left\|\mathbf{v} - \mathbf{z}_{N+1}\right\|_2^2 = \Pi_{\mathbb{R}_+}\left(\mathbf{v}\right), \tag{18}$$

where $\Pi_{\mathbb{R}_+}\left(\cdot\right)$ is the element-wise projection operator onto the convex set $\mathbb{R}_+$, see [12]. Again, replacing the free variable $\mathbf{v}$ with the corresponding value for $\mathbf{x} + \mathbf{u}_{N+2}$ in line 3 of Alg. 1 results in the specific algorithm step.

### 3.4. Total-Variation Step in Algorithm 1

The subproblem in line 5 and line 6 of Alg. 1 also have the form of a well-known proxoimal operator. Specifically, these algorithms steps can be defined as the following Shrinkage operations [12] using the free substitution variable $\mathbf{v}$

$$\begin{aligned}
\underset{\mathbf{z}_{N+2}}{\operatorname{argmin}} \; \lambda\|\mathbf{z}_{N+2}\|_1 + \frac{\tau}{2}\left\|\mathbf{v} - \mathbf{z}_{N+2}\right\|_2^2 &= \max\left(1 - \frac{\tau\lambda}{|\mathbf{z}_{N+2}|}, 0\right) \odot \mathbf{z}_{N+2} \\
\underset{\mathbf{z}_{N+3}}{\operatorname{argmin}} \; \lambda\|\mathbf{z}_{N+3}\|_1 + \frac{\tau}{2}\left\|\mathbf{v} - \mathbf{z}_{N+3}\right\|_2^2 &= \max\left(1 - \frac{\tau\lambda}{|\mathbf{z}_{N+3}|}, 0\right) \odot \mathbf{z}_{N+3}
\end{aligned} \tag{19}$$

which implements an anisotropic Total Variation regularizer following [3, 12]. The specific algorithm updates follow from replacing $\mathbf{v}$ with the corresponding iteration value for $\mathbf{x} + \mathbf{u}_{N+2}$ and $\mathbf{x} + \mathbf{u}_{N+3}$, respectively.

## 4. Synthetic Rendering Details

The success of deep learning relies heavily on the amount of training data. Learning NLOS imaging requires us to provide thousands of indirect reflection maps together with their ground truth orthogonal projection on the wall area, which we use as latent parametrization. Capturing such a dataset experimentally would be extremely challenging and requires placing thousands of different objects in different locations and orientations, projecting different light beams and capturing the corresponding ground truth image from the center of the wall. To avoid this immense experimental effort, we propose to synthesize training images instead of capturing them.

Recently, synthesizing training data has been shown to be an efficient way to boost deep learning training. For example, [13, 4] show that the models trained with synthetic data can perform as good as real data on the tasks of view estimation or pose estimation. Moreover, ShapeNet[15] supplies thousands of 3D models for different categories, which covers almost many common objects in real world. With such a rich 3D model dataset in hand, we can synthesize virtually infinite indirect training variations for different shapes, textures and viewpoints.

However, our render task is very different from [13, 4]. While the approaches rely on direct illumination rendering using by traditional OpenGL rendering pipeline, we have to generate indirect reflection measurements with third-bounce global light transport for each measurement pixel. As shown in Sec. 3 of the main document, each pixel in the indirect reflection map is the radiance integral over the hemisphere around this pixel. This integral can be handled by ray tracing, where one simulates paths from the camera to the object. To sample this large space of paths, ray tracing requires us to calculate millions of rays, which is computationally costly.

**Render Pipeline** With the goal of making synthesizing large training datasets of indirect observations feasible, we propose the following render pipeline. As shown in Fig. 3, consider a pixel $c$ on the image plane, which receives lights from potentially all direction of the surface-hemisphere centered on this point, such as $s_1$, $s_2$ and many other directions on the hidden object. The traditional OpenGL pipeline cannot handle such a case, since it assumes pixel receives light that comes from only one direction.

To estimate this integral, we sample the incident radiance contributions at $c$ on the hemisphere. Hence, we can render each view direction using hardware-accelerated OpenGL in microseconds. Finally, accumulating all rendered images over the hemisphere, weighted by the according diffuse wall BRDF, yields the resulting indirect reflection map. Our data generation

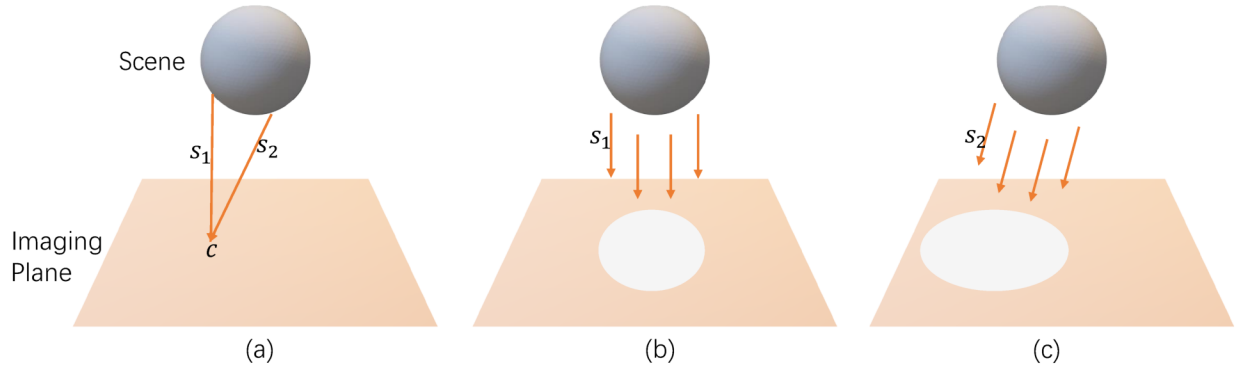


Figure 3. (a) a pixel $c$ in the indirect reflection map will be influenced by lights from all the directions, which makes this pixel an integral and hard to be rendered in traditional OpenGL pipeline. (b), (c) We decompose lights into different directions and use orthographics projection to render them separately. (c) shows an example of non-perpendicular direction rendering, where the sphere object is projected to an ellipse. Finally we sum over all the rendering and get the simulated indirect reflection map.

pipeline adopts orthographic projections, and we comfortably change projection direction in an OpenGL vertex shader. Fig. 3 (b) and (c) show an example. For case $s_1$, its direction is perpendicular to the image plane, thus we can directly render it with orthographic projection. For $s_2$, we need to change the projection direction. Now, the spherical object will be rendered into an ellipse with a non-perpendicular light direction. $c$ will receive light path contributions from all directions, and hence we render image from all the directions on the hemisphere. Thus we uniformly sample $100 \times 100$ different light directions on the hemisphere. Fig. 4 shows an indirect rendering example of an MNIST character as albedo on a hidden planar surface.

## 5. Extended Results & Comparisons

### 5.1. Additional Experimental Results for Planar Recovery Method

Fig. 6-9 show additional experimental results for the special case of planar scene surfaces. The visualizations in the first row show the proposed optimization approach described in Sec. 4 of the main draft. The second row visualizes results obtained from the backprojection method described in [14] applied on the steady-state measurement, i.e. without temporal information. The last row shows the result of the total-variation-regularized linear inverse method from [8], which also achieves results on-par with [11] when applied on the same diffuse measurements. Both comparison approaches assume isotropic scattering and temporal light transport sampling, and hence fail to recover detailed albedo reconstructions, in contrast to the proposed reconstruction method. However, although no temporal information is provided, both compared methods allow to recover the rough location of the scene object from the diffuse indirect reflection component.

The results in Fig. 6-9 validate the proposed method experimentally for a variety of specular surface materials, including conventional painted surfaces to even retroreflective street signs, which are engineered to maximize the retroreflective BRDF component. Our method recovers high-quality specular albedo and planar geometry while running at around two seconds of capture and optimization runtime.

We measure an average position error bounds of [-5,+5] cm in lateral position and [-5,+5] ° in plane normal angle, averaged over all planar experimental scene. We experimented with a variety of different surface types, including almost diffuse, painted surfaces, engineering-grade retroreflextive materials, diamond-grade retroreflective materials, ceramic materials, wrapped reflective materials and (dominatingly) diffuse cardboard. The proposed method failed only for dominatingly diffuse objects, and provided even good performance for minimal specular lobes, as in the case of the retroreflective objects shown in this work. We achieve between 3 - 10 feature matches, depending on the surface material properties. Fig. 5 shows a matching example. Although the specular part is very faint, vanilla SIFT features with normalization allow to find sufficient matches across the $5 \times 5$ measurements. In practice, we were able to reduce the sampling density to $3 \times 3$ without significant loss in reconstruction quality.

### 5.2. Additional Synthetic Results

Figure 10 provides more reconstruction examples on synthetic dataset. The first two columns are two examples of synthetic indirect reflection maps. The third row is the ground truth orthographic rendering with the camera position in the center of the wall, and the fourth column is our prediction. For each category, we display 2 examples. We train our network with Tensorflow[1] framework and adopt Adam[9] algorithm as the optimizer. The model is trained with 5e-5 learning rate, where

it decays 1e-5 every 3 epoches. Trained on a NVIDIA GTX 1080Ti GPU card, it will train 15 epoches in total and takes around 10 hours to converge.

### 5.3. Additional Experimental Results for Learned Recovery Method

Fig. 11 and Fig . 12 show additional experimental results for various object classes. In the first two columns, we show the top and side view of the hidden object. The third and fourth columns show captured indirect reflection measurements. We show our reconstruction results in fifth column. The sixth column visualizes results obtained from the backprojection method described in [14] (without filtering) applied on the steady-state measurement, i.e. without temporal information. The last column shows the result of the total-variation-regularized linear inverse method from [8].

The reconstruction results validate that the proposed learned model, which is trained on synthetic data, successfully generalizes to challenging NLOS recovery tasks on real captures. For large or sparse objects, such as the character cutout and the chair or doll examples, we predict their reasonable shape. Note that the object surface of the character scenes is completely diffuse, while the doll objects contain slight specular lobes. While these objects are well reconstructed, the reconstructions for scenes with complicated geometry and reflectance suffer from artefacts, such as the car and chair examples. The method matches the simulation results in these cases, and the predictions recover accurate position and only rough shape, sufficient to localize the object. The compared backprojection and linear inverse method assume high temporal resolution measurements and isotropic reflectance and hence fail in these scenarios.

## References


[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 7

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found.&Trends® in Machine Learning*, 3(1):1–122, 2011. 4, 5, 6

[3] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging and Vision*, 40(1):120–145, 2011. 5, 6

[4] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *3D Vision (3DV)*, 2016. 6

[5] F.-X. Dupe, M. Fadili, and J.-L. Starck. Inverse problems with poisson noise: Primal and primal-dual splitting. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1901–1904, 2011. 5, 6

[6] M. A. Figueiredo and J. M. Bioucas-Dias. Restoration of poissonian images using alternating direction optimization. *IEEE transactions on Image Processing*, 19(12):3133–3145, 2010. 4, 5

[7] F. Heide, S. Diamond, M. Niessner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein. Proximal: Efficient image optimization using proximal algorithms. *ACM Trans. Graph. (SIGGRAPH)*, 35(4):84:1–84:15, 2016. 5

[8] F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (ToG)*, 32(4):45, 2013. 5, 7, 8, 11, 12, 13, 14, 16, 17

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 7

[10] M. O'Toole, F. Heide, D. B. Lindell, K. Zang, S. Diamond, and G. Wetzstein. Reconstructing transient images from single-photon sensors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2289–2297, July 2017. 5

[11] M. O'Toole, D. B. Lindell, and G. Wetzstein. Confocal non-line-of-sight imaging based on the light cone transform. *Nature*, pages 338–341, 2018. 5, 7

[12] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, pages 1–96, 2013. 5, 6

[13] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 6

[14] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, 3:745, 2012. 7, 8, 11, 12, 13, 14, 16, 17

[15] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6, 15

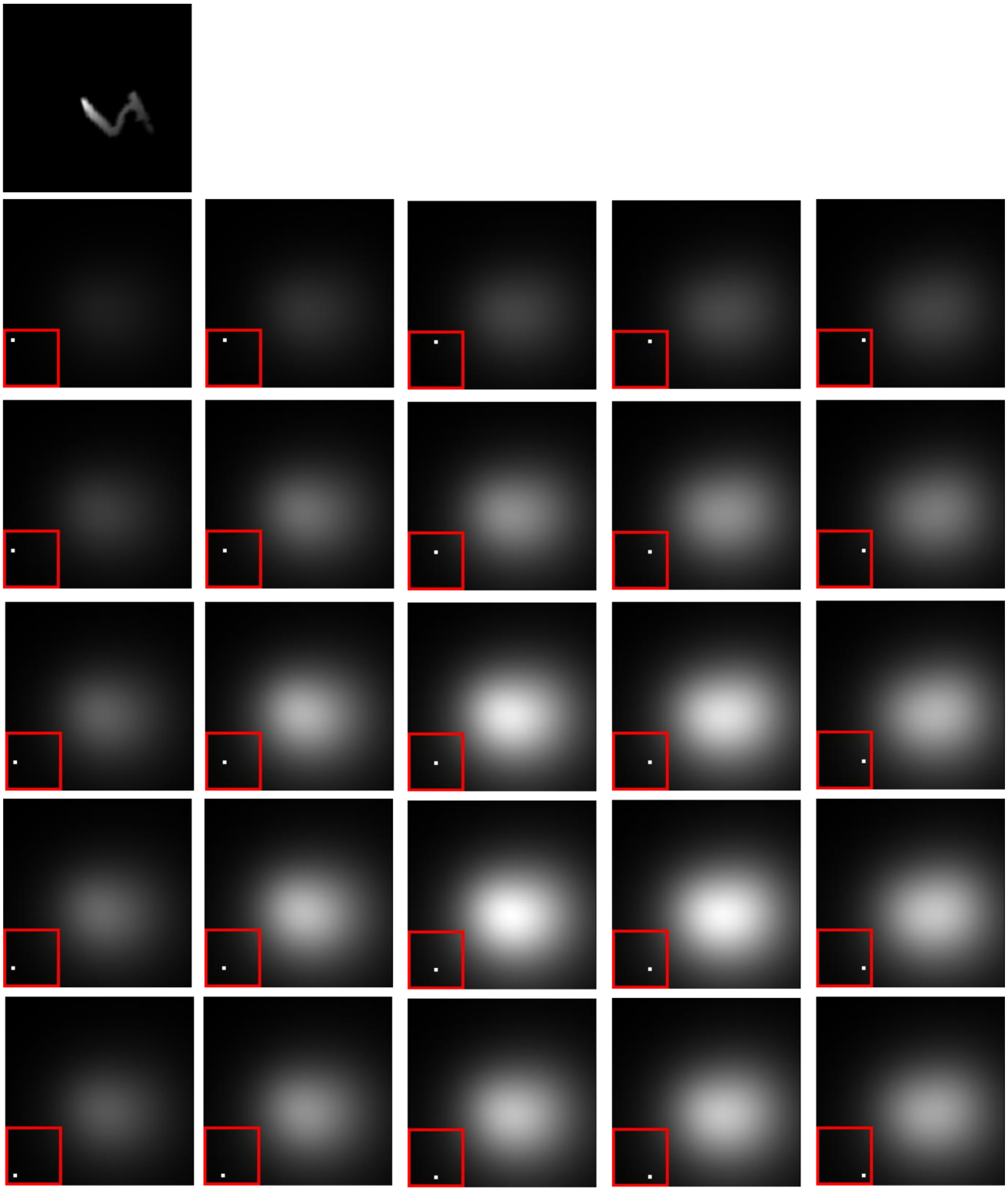

Figure 4. Render examples. The top image shows an orthographic rendering of a hidden planar object with MNIST character texture. Bottom $5 \times x5$ images are synthetic indirect reflection maps with a $5 \times 5$ grid of virtual light positions.

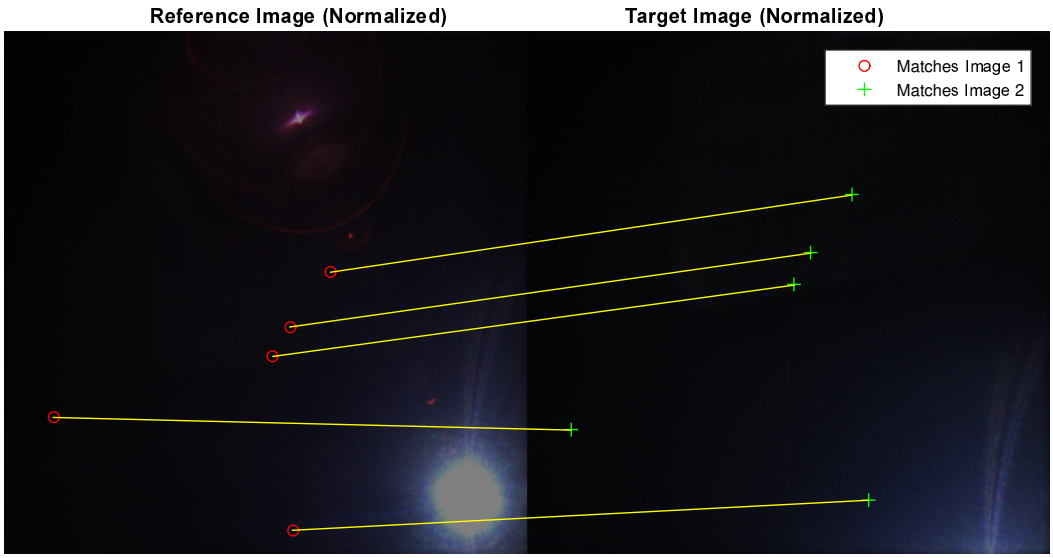


Figure 5. Example matches between a reference image and another image. We have cropped and normalized the captures to make the faint indirect reflection better visible. While two erroneous matches are found, also three good matches are found for this challenging low-flux feature matching process. Note that this particular sign did not exhibit significant specular blur, aiding the reconstruction and matching. While some matches can be incorrect, note that the $5 \times 5$ sampling provides 24 potential image pairs for matching.

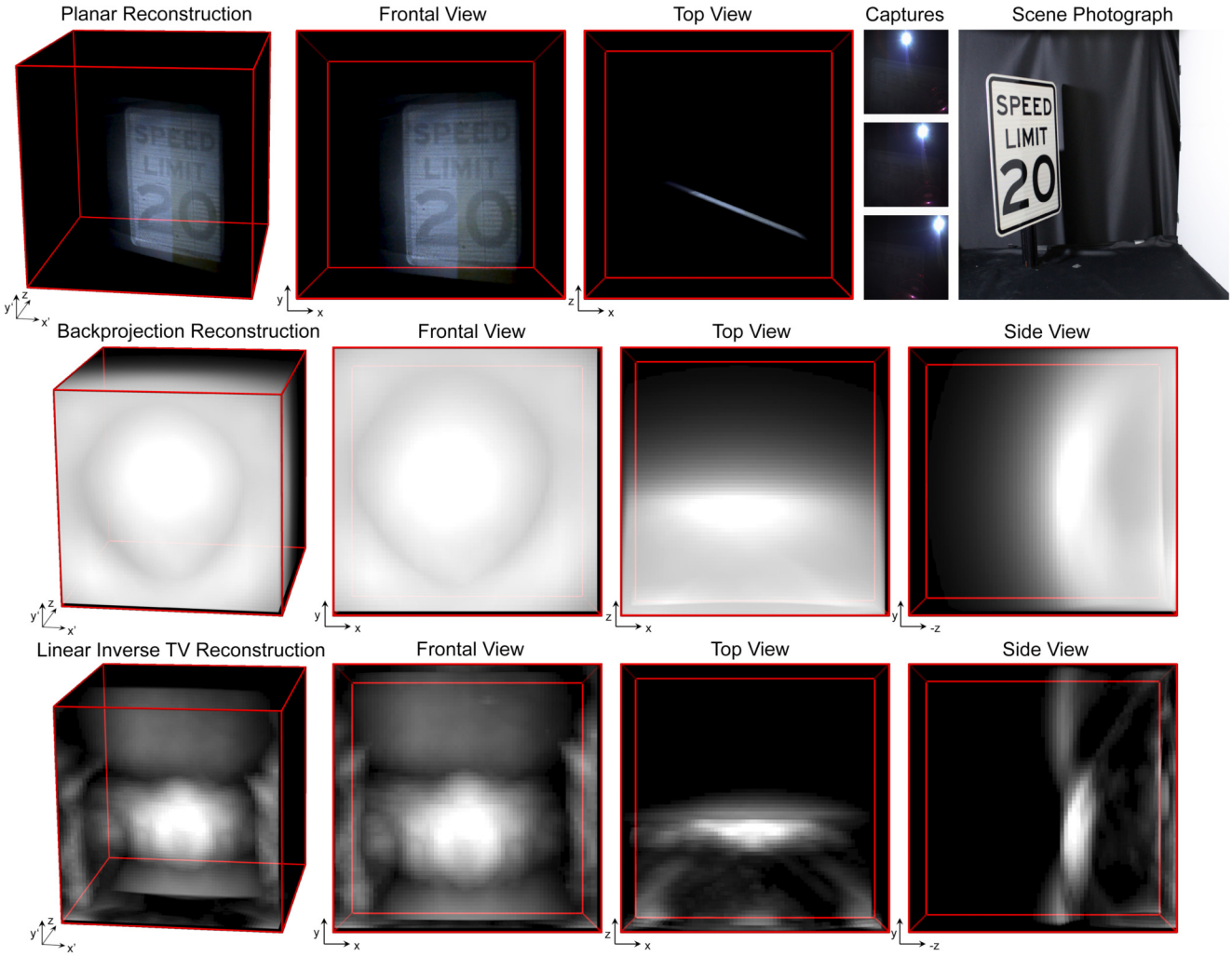


Figure 6. Additional experimental geometry and albedo reconstructions for the special case of planar objects, captured with the prototype from Sec. 7 from the main draft. In the first row, we show reconstruction results for a retroreflective street sign which, although designed to be retroreflective, contains faint specular components visible in the measurements. The second row visualizes results obtained from the backprojection method described in [14] applied on the steady-state measurement, i.e. without temporal information. The last row shows the result of the total-variation-regularized linear inverse method from [8].

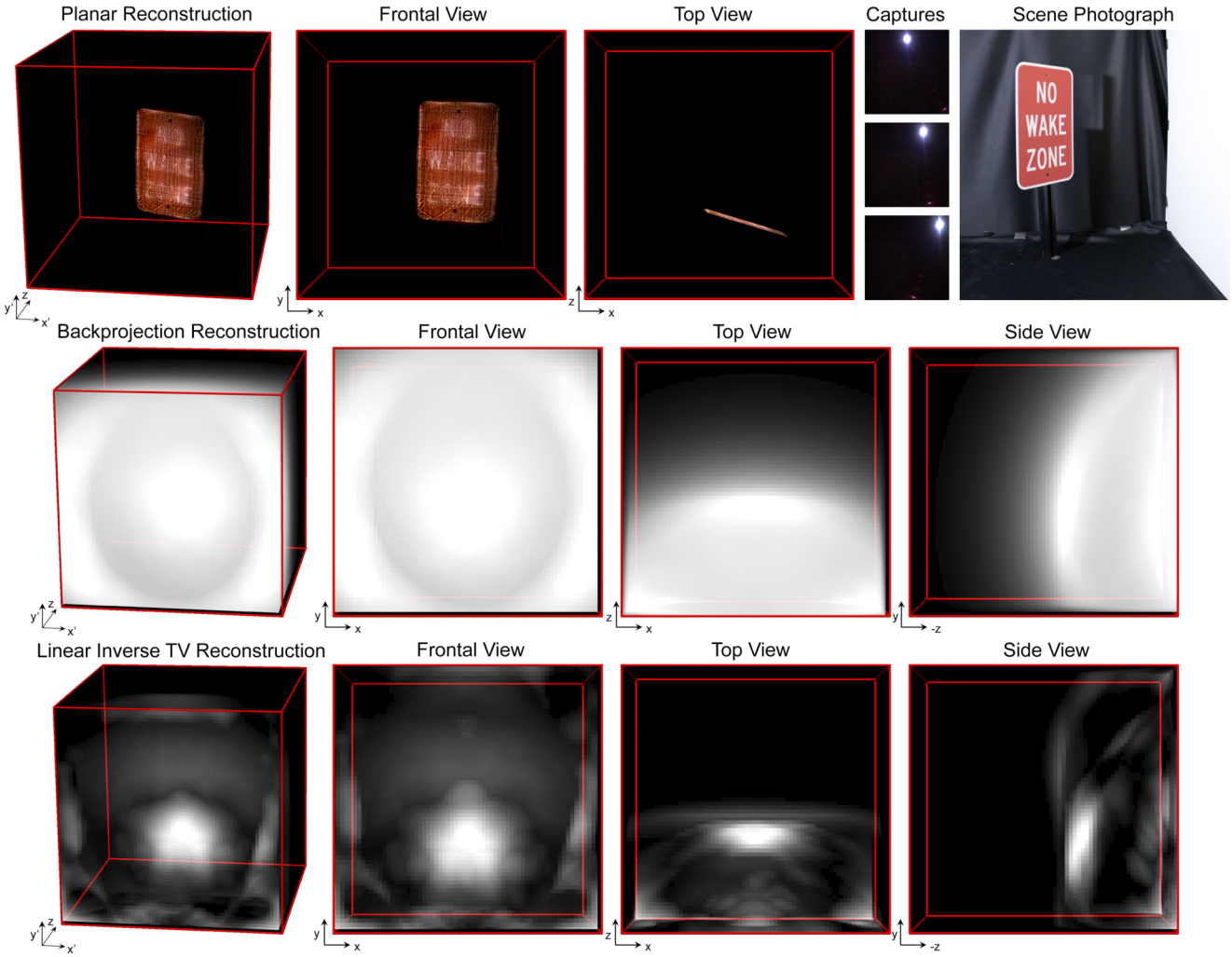


Figure 7. Additional experimental geometry and albedo reconstructions for the special case of planar objects, captured with the prototype from Sec. 7 from the main draft. In the first row, we show reconstruction results for an engineering grade street sign. The second row visualizes results obtained from the backprojection method described in [14] applied on the steady-state measurement, i.e. without temporal information. The last row shows the result of the total-variation-regularized linear inverse method from [8].

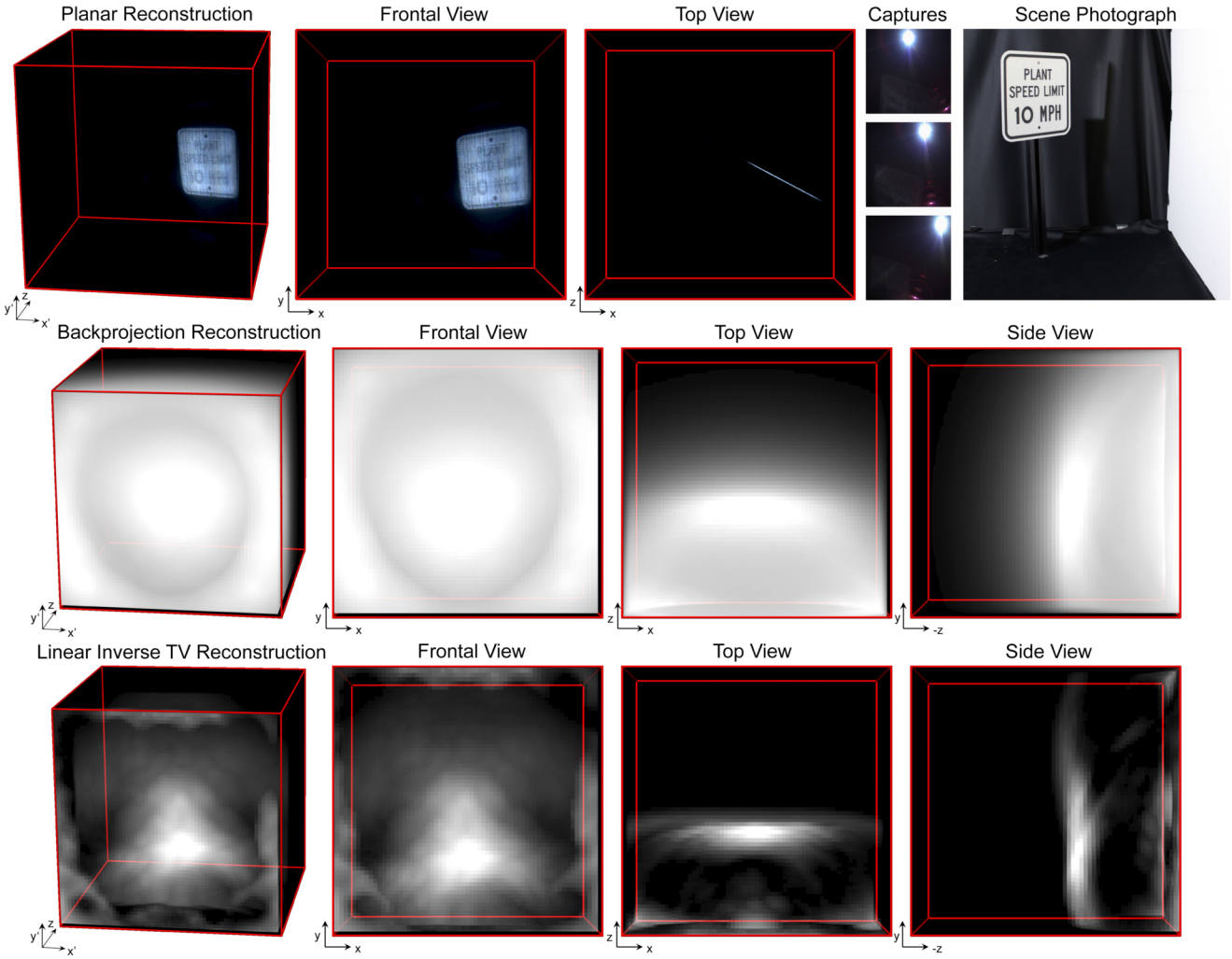


Figure 8. Additional experimental geometry and albedo reconstructions for the special case of planar objects, captured with the prototype from Sec. 7 from the main draft. In the first row, we show reconstruction results for a diamond grade street sign which. The second row visualizes results obtained from the backprojection method described in [14] applied on the steady-state measurement, i.e. without temporal information. The last row shows the result of the total-variation-regularized linear inverse method from [8].

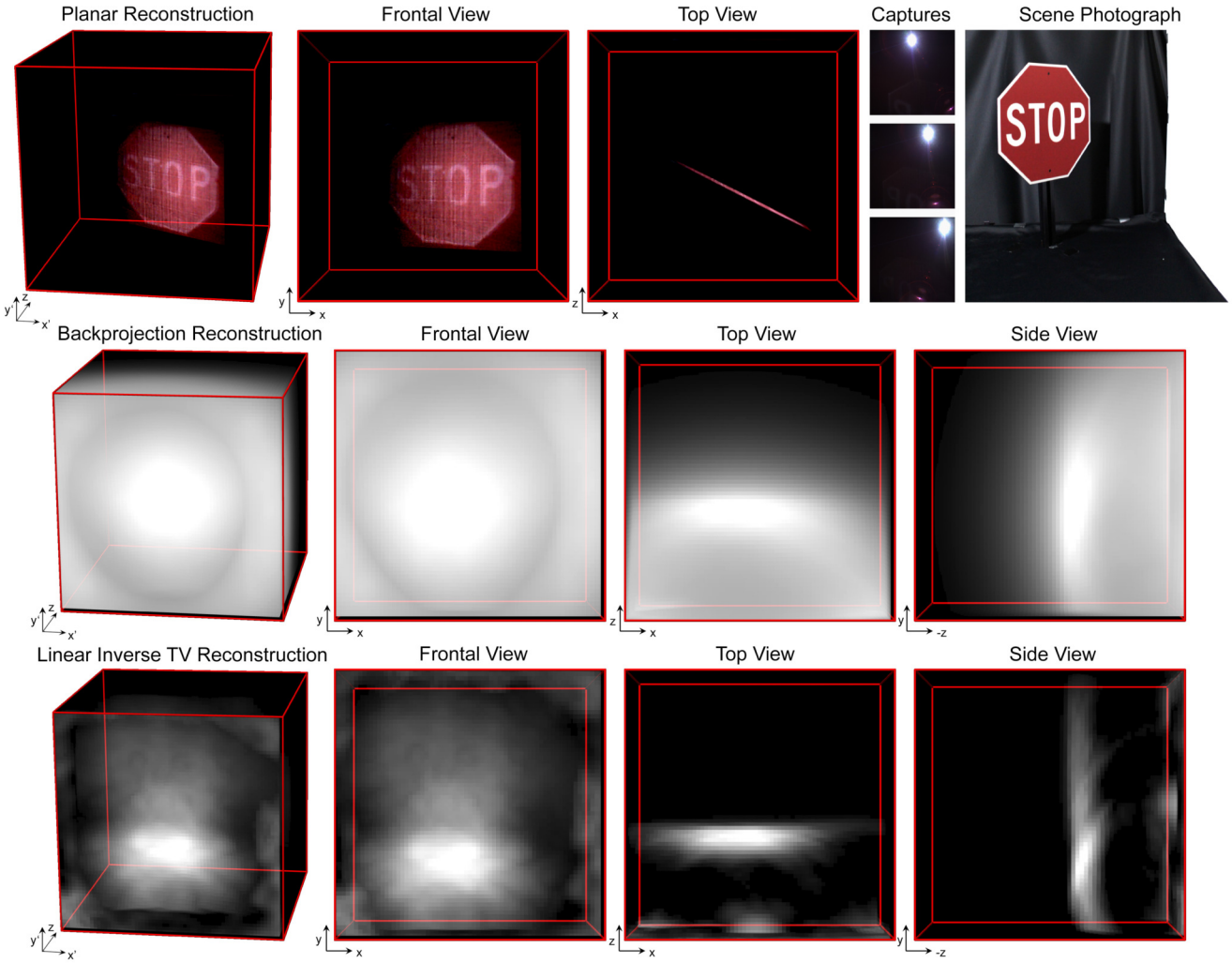


Figure 9. Additional experimental geometry and albedo reconstructions for the special case of planar objects, captured with the prototype from Sec. 7 from the main draft. In the first row, we show reconstruction results for a conventional painted street sign. The second row visualizes results obtained from the backprojection method described in [14] applied on the steady-state measurement, i.e. without temporal information. The last row shows the result of the total-variation-regularized linear inverse method from [8].

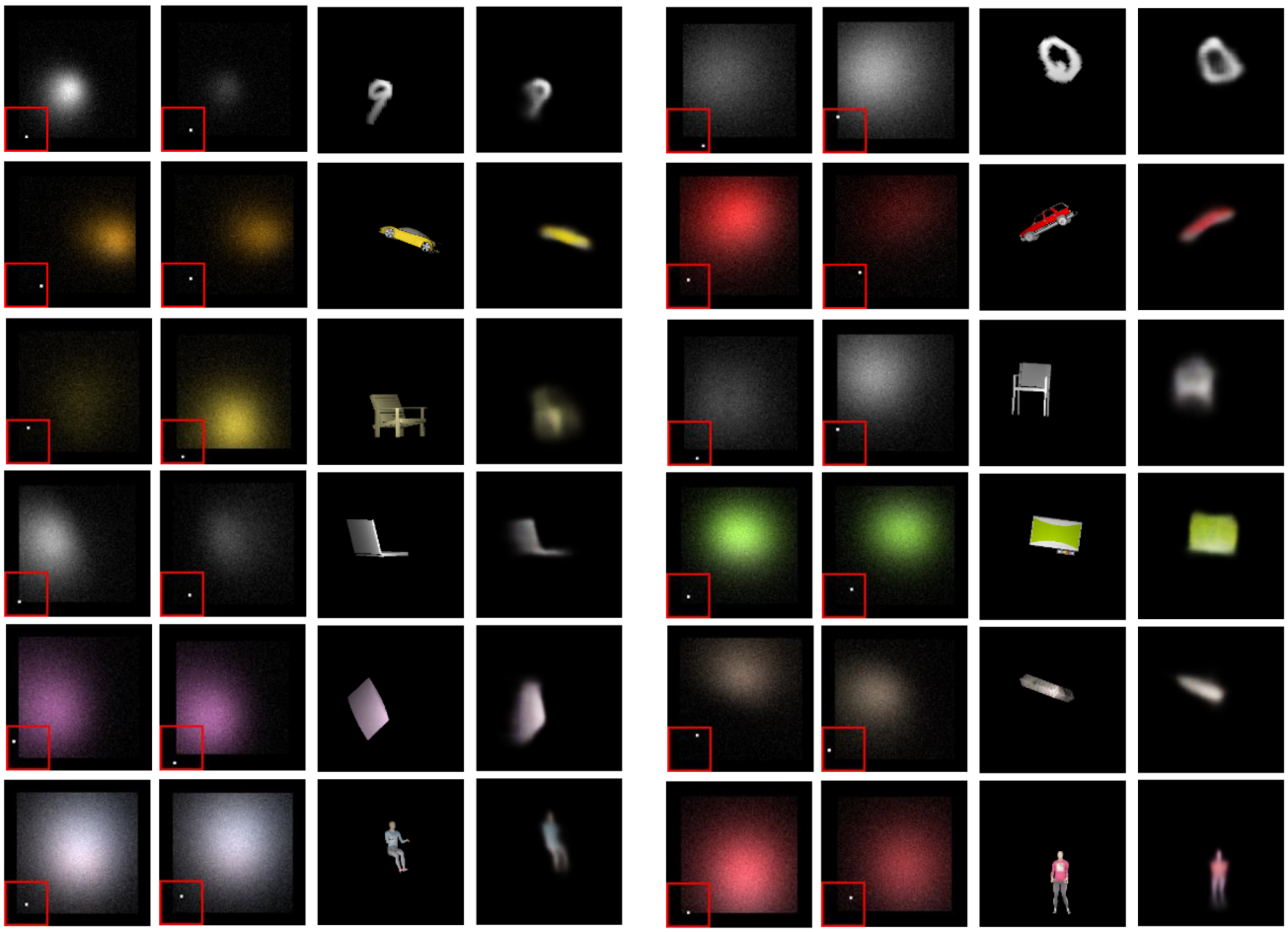

Figure 10. Additional synthetic reconstructions for MNIST digit numbers, ShapeNet[15] classes and human models. Six rows from top to bottom are planar MNIST characters, cars, chairs, laptops, pillows and human models, respectively. For each class we show 2 examples. The first two columns are synthetic indirect reflection maps with added noise. The third column is the object appearance ground truth while the forth column is our prediction.

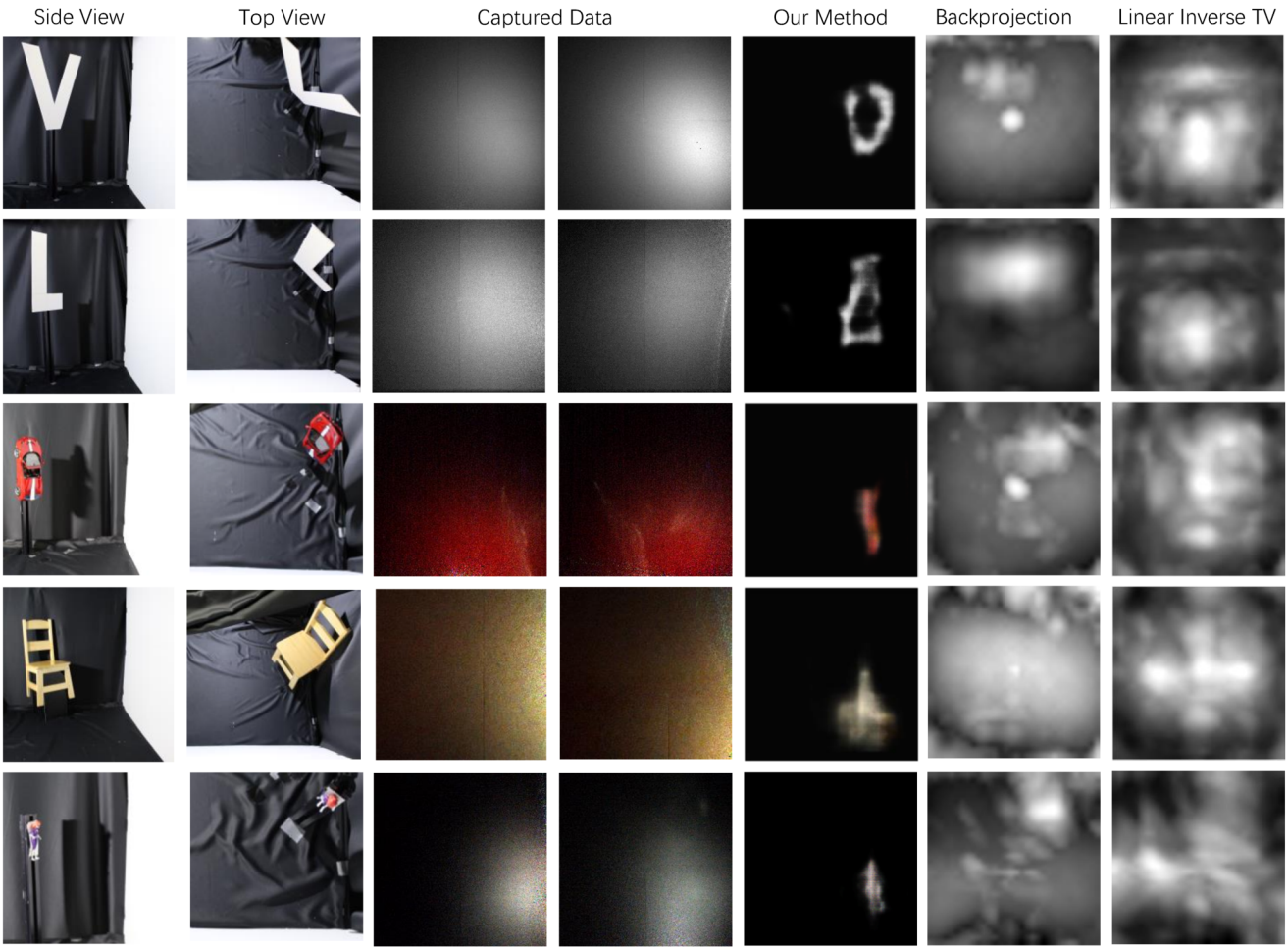


Figure 11. Additional experimental learned NLOS reconstructions for various objects, captured with the prototype from Sec. 7 from the main draft. In the first two columns, we show side view and top view of the object. The third and forth columns are two examples of captured data. We also compare our approach against backprojection [14] (without filtering) and [8], where the fifth, sixth and seventh column are prediction of our method, [14] and [8], respectively. For all the methods, we show the orthogonal projection on the wall.

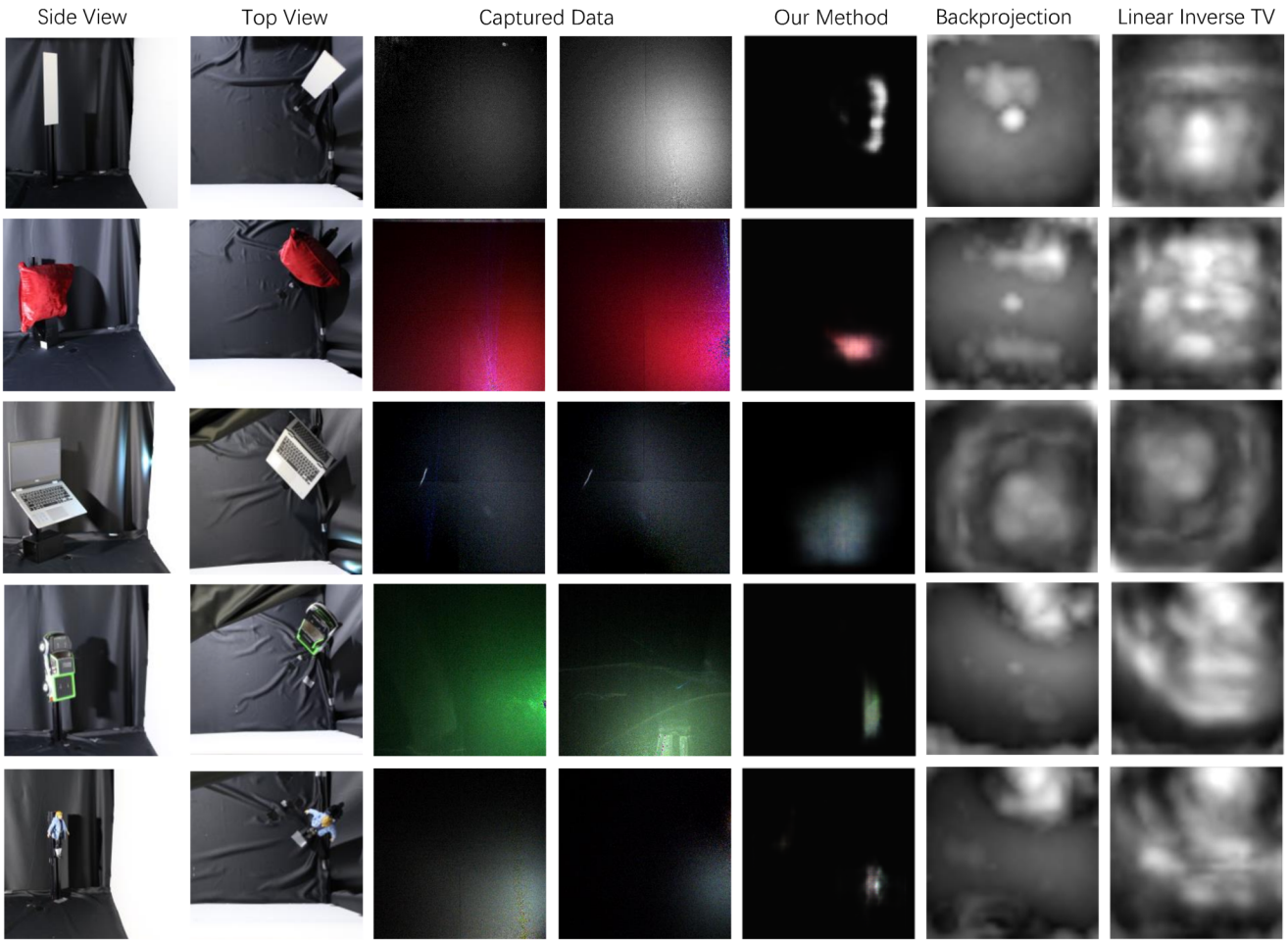


Figure 12. Additional experimental learned NLOS reconstructions for various objects, captured with the prototype from Sec. 7 from the main draft. In the first two columns, we show side view and top view of the object. The third and forth columns are two examples of captured data. We also compare our approach against backprojection [14] (without filtering) and [8], where the fifth, sixth and seventh column are prediction of our method, [14] and [8], respectively. For all the methods, we show the orthogonal projection on the wall.