

# InvRGB+L: Inverse Rendering of Complex Scenes with Unified Color and LiDAR Reflectance Modeling

## Supplementary Material

### 1. Custom Data Collection

We recorded data ourselves using two LiDAR-camera systems (shown in Fig. 1) to enable our outdoor quantitative relighting evaluation and indoor albedo-reflectance experiments. In both cases, we used targetless LiDAR-camera calibration [4] to obtain the coordinate transform between the LiDAR and camera.

**Outdoor** We recorded data from the Polaris Gem e4, a street-legal, four-seater vehicle outfitted with RTK GPS, an Ouster OS1-128 LiDAR, and an Oak-D LR stereo camera. The experiments were carried out in a shared testing track facility with a secure testbed area. A designated safety driver and safety lookout were present at all times. We collected the dataset by manually driving the vehicle along the same trajectory/scene at different times throughout the day.

**Indoor** We used an AgileX Ranger Mini 2 mobile robot platform with a Hesai FT120 solid-state LiDAR and Realsense D455 depth camera mounted on top. We recorded experiments by teleoperating the robot inside an academic building.



Figure 1. The two LiDAR-camera systems used for data collection. The left is for outdoor, and the right is for indoor.

### 2. LiDAR Intensity Reveals Specularity

We verify that LiDAR intensity can indicate surface specularity by collecting real-world data of a diffuse wall and a specular whiteboard. In Fig. 2, the specular whiteboard has a high intensity only around the center of the image where the LiDAR beams are parallel to the surface normal of the whiteboard. Since it is a specular surface, those parallel LiDAR beams are reflected back at the same angle, straight into the sensor, and the LiDAR receiver gets the strongest signal/highest intensity in that region. The diffuse wall, however, shows no major intensity difference and

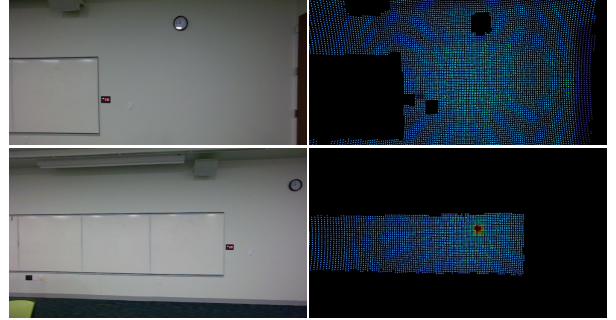


Figure 2. RGB and masked lidar intensity for a diffuse wall (top) and a specular whiteboard (bottom).

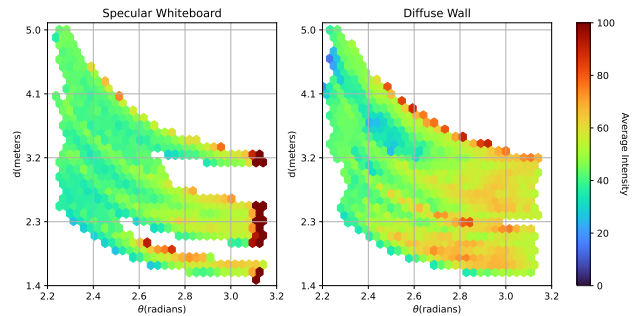


Figure 3. LiDAR intensity is visualized for two different objects: a specular whiteboard and a diffuse wall. The x-axis,  $\theta$ , is the angle between a LiDAR ray and the corresponding surface normal. The y-axis,  $d$ , is the distance to the LiDAR point. Each hexbin represents the average LiDAR intensity among all the LiDAR points within that bin.

is roughly uniform across all the LiDAR points since it reflects light in all directions.

In Fig. 3, we show this continues to hold true at varying distances and reflectance angles. We recorded a sequence of data scanning both objects and accumulated the intensities for points corresponding to each one. For the specular whiteboard, the highest intensity LiDAR points are clustered around  $\theta = 3.14$ , when the LiDAR ray and surface normal are parallel. But for the diffuse wall, the intensity is roughly spread out as expected. This shows how LiDAR intensity can be a valuable cue to determine specularity.

### 3. Physics-based LiDAR Reflectance Model

Here, we provide the mathematical derivation for the specular term of the LiDAR Reflectance Model. The Cook-Torrance BRDF model [2] for the specular component is

given by:

$$\begin{aligned} f_s(\omega_i, \omega_o) &= \frac{F(\omega_i) G(\omega_i, \omega_o) D(\mathbf{h})}{4(\omega_i \cdot \mathbf{n})(\omega_o \cdot \mathbf{n})} \\ &= \frac{F(\omega_i) G(\omega_i, \omega_o) D(\mathbf{h})}{4 \cos^2 \theta}, \end{aligned}$$

where:

- $D(\mathbf{h})$  is the microfacet distribution function, modeling the distribution of surface normals, with  $\mathbf{h}$  being the half-angle vector.
- $G(\omega_i, \omega_o)$  is the geometry term, accounting for masking and shadowing of microfacets.
- $F(\omega_i)$  is the Fresnel term, which models reflectance based on the viewing angle.

For the LiDAR lighting model, where  $\omega_i = \omega_o$ , the half-angle vector simplifies to:

$$\mathbf{h} = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|} = \omega_o.$$

The microfacet distribution function is commonly modeled using the GGX distribution:

$$\begin{aligned} D(\mathbf{h}) &= \frac{\alpha^2}{\pi ((\mathbf{h} \cdot \mathbf{n})^2 (\alpha^2 - 1) + 1)^2} \\ &= \frac{\alpha^2}{\pi (\cos^2 \theta (\alpha^2 - 1) + 1)^2}, \end{aligned}$$

where  $\alpha$  represents the surface roughness.

For the Fresnel term, we use:

$$F(\omega_i) = F_0 + (1 - F_0)(1 - (\omega_i \cdot \mathbf{h}))^5 = F_0.$$

For the geometry term, following the Cook-Torrance BRDF model, it can be calculated as:

$$\begin{aligned} G &= \min \left( 1, \frac{2(\omega_i \cdot \mathbf{n})(\mathbf{n} \cdot \mathbf{h})}{(\omega_o \cdot \mathbf{h})}, \frac{2(\omega_o \cdot \mathbf{n})(\mathbf{n} \cdot \mathbf{h})}{(\omega_i \cdot \mathbf{h})} \right) \\ &= \min(1, 2 \cos^2 \theta). \end{aligned}$$

Thus, the final specular term is:

$$\begin{aligned} f_s(\omega_i, \omega_o) &= \frac{F(\omega_i) G(\omega_i, \omega_o) D(\mathbf{h})}{4 \cos^2 \theta} \\ &= \frac{F_0 \alpha^2 \min(1, 2 \cos^2 \theta)}{4 \pi \cos^2 \theta (\cos^2 \theta (\alpha^2 - 1) + 1)^2}. \end{aligned}$$

## 4. Implementation Details

**Data Preprocessing** For image inputs, we preprocess each frame and acquire diffusion-based priors for materials and normals using RGB $\leftrightarrow$ X [5] and GeoWizard [3]. For the indoor scene, we obtain an additional lighting mask  $\mathbf{M}_{\text{light}}$  by filtering the luminance. This mask is used to exclude unreliable regions from the rendering loss, where high-intensity lighting makes it unreliable to estimate albedo.

For LiDAR sequences, we project the LiDAR points into image space using the camera-LiDAR pose transformation, resulting in a sparse LiDAR intensity map. The intensity values are then normalized to the range [0, 1]. Furthermore, the Waymo dataset provides the ground truth object poses for each object, which are used as  $\mathbf{T}_k(t)$  to transform dynamic nodes into the world coordinate system.

**Method Details** We develop our framework based on OmniRe [1], a 3D-GS framework designed for driving scene. We initialize the means of the background 3D-GS with LiDAR points. Specifically, we set the maximum point number to  $8 \times 10^5$ . If the number of LiDAR points exceeds this limit, we randomly sample points. For rigid nodes, we randomly sample 5,000 points for initialization within the 3D bounding boxes.

For the camera rendering process, we adopt Monte Carlo ray tracing. For each Gaussian primitive  $g$ , we generate  $M$  incident ray directions using Fibonacci sphere sampling based on the normal direction. During training, we set  $M = 16$ , while for inference, we use  $M = 128$ .

**Optimization** The model is trained for 30,000 iterations with a single NVIDIA A100 GPU. It takes approximately 2-3 hours of training for each scene. The learning rate is set to  $1 \times 10^{-5}$ . We adopt a two-stage training procedure: in the first stage, which consists of the first 15,000 iterations, we follow the general 3D-GS split replication approach, and all intrinsic properties are optimized simultaneously. After completing the first stage, the LiDAR albedo is processed into masks, and we stop duplicating and deleting 3D-GS nodes. In the next 15,000 iterations, all intrinsic properties except for LiDAR albedo and RGB albedo are fixed, and lighting conditions are also optimized. The total loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \lambda_1 \mathcal{L}_{\text{lidar}} + \lambda_2 \mathcal{L}_{\text{rgb}} + \lambda_3 \mathcal{L}_{\text{nor}} + \lambda_4 \mathcal{L}_{\text{mat}} \\ &\quad + \lambda_5 \mathcal{L}_{\text{rgb} \rightarrow \text{lidar}} + \lambda_6 \mathcal{L}_{\text{lidar} \rightarrow \text{rgb}}, \end{aligned} \quad (1)$$

where  $\lambda_i$  is the loss weight for each term. Specifically, we set  $\lambda_1 = \lambda_2 = 1$ ,  $\lambda_3 = \lambda_4 = 0.1$ , and  $\lambda_5 = \lambda_6 = 0.05$ .

**Application** For object insertion, we transfer the trained dynamic nodes from the Waymo dataset into new scenes by

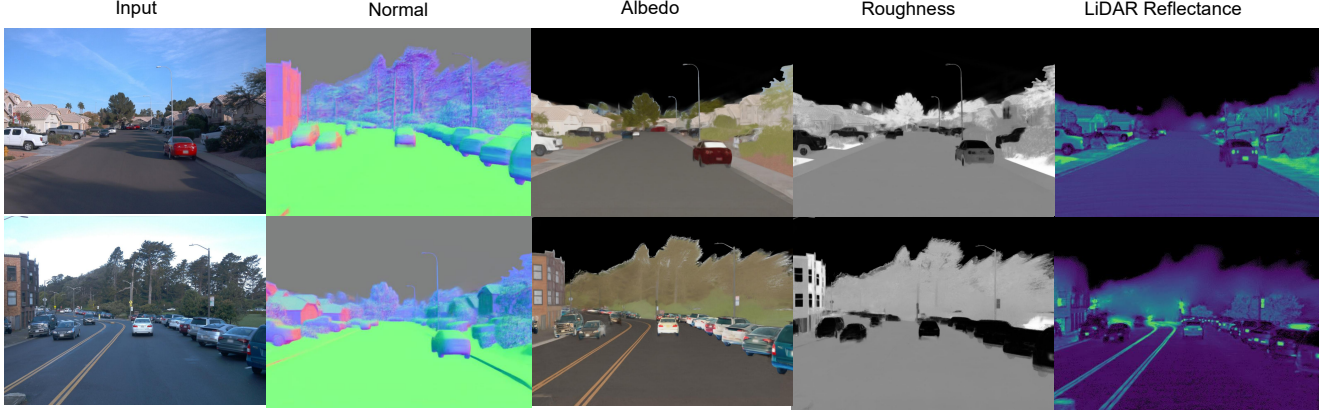


Figure 4. More results for inverse rendering.

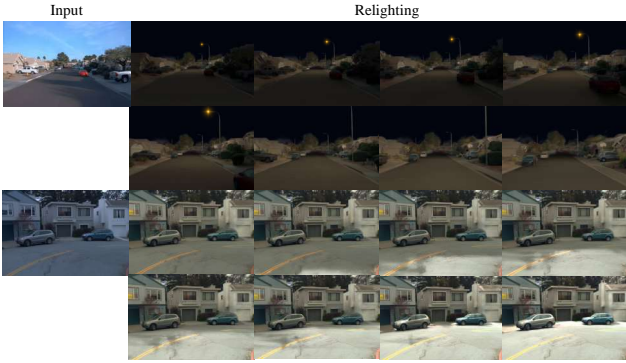


Figure 5. Relighting results on video sequences.

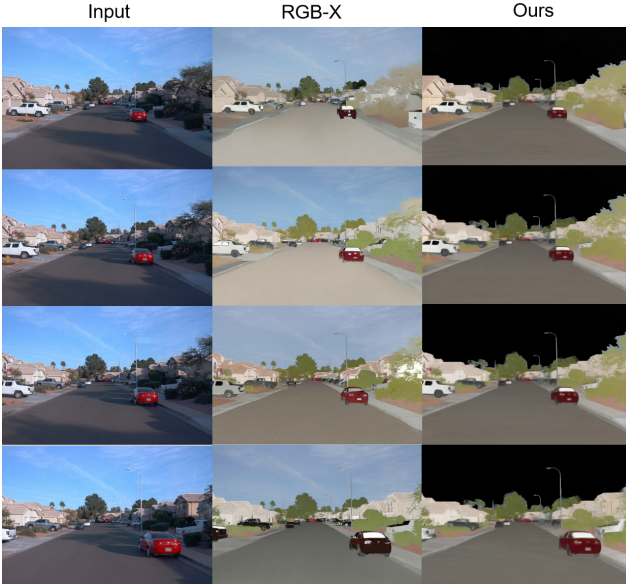


Figure 6. Comparison for albedo estimation with RGB $\leftrightarrow$ X.

directly loading the corresponding checkpoints. Since the dynamic nodes retain their intrinsic properties, a relit re-



Figure 7. Ablation study on diffusion prior.

sult can be directly obtained through our camera rendering process. For nighttime simulation, we remove the sunlight representation and set the sky dome lighting to a small constant intensity. Additionally, we use a spotlight to model the headlights and streetlights, with its center positioned at the camera origin or the light pole. The light intensity decreases with the square of the distance from the spotlight center to the Gaussian’s means.

## 5. More Qualitative Results

**More Inverse Rendering** Fig. 4 presents additional inverse rendering results, including normal, albedo, roughness, and LiDAR reflectance. As shown in the figure, the LiDAR and RGB albedo are consistent, and we can successfully disentangle shadows from the albedo.

**More Relighting Results** We perform relighting on two sequences: the first simulates nighttime conditions, and the second continuously changes the sun direction. We present 8 frames of each video in Fig. 5.

**Comparison with Diffusion Prior** Fig. 6 compares the albedo estimation results of our method against RGB $\leftrightarrow$ X on an image sequence. The albedo prior from RGB $\leftrightarrow$ X exhibits significant temporal inconsistency due to the inherent limitations of the monocular diffusion model. In contrast, our framework achieves time-consistent albedo estimation,

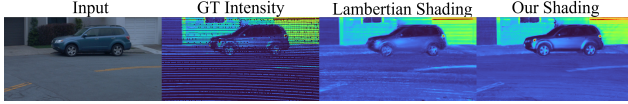


Figure 8. Ablation study on LiDAR reflectance modeling.

Method	Intensity-RMSE ↓
Lambertian	0.0493
Ours (PBR)	<b>0.0470</b>

Table 1. Ablation study on LiDAR reflectance modeling.

highlighting the advantages of physically based optimization.

**Ablation on Diffusion Prior** We conduct an ablation study to assess the role of the diffusion prior. As shown in Fig. 7, incorporating the diffusion prior produces smoother albedo estimates and reduces shadow ambiguity (e.g., on the trees), highlighting its effectiveness.

**Ablation on LiDAR Reflectance Modeling** Tab. 1 and Fig. 8 provide ablation studies of LiDAR simulation on a scene. Our reflectance model faithfully captures the specular highlights in the GT intensity—e.g., around the car’s front light and wheel arch—whereas the Lambertian model produces overly diffuse, physically implausible shading.

## References

- [1] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 2
- [2] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982. 1
- [3] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision*, pages 241–258. Springer, 2024. 2
- [4] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023-5-29. 1
- [5] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb-x: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, page 1–11. ACM, 2024. 2