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

Anonymous ICCV submission

Paper ID 4741

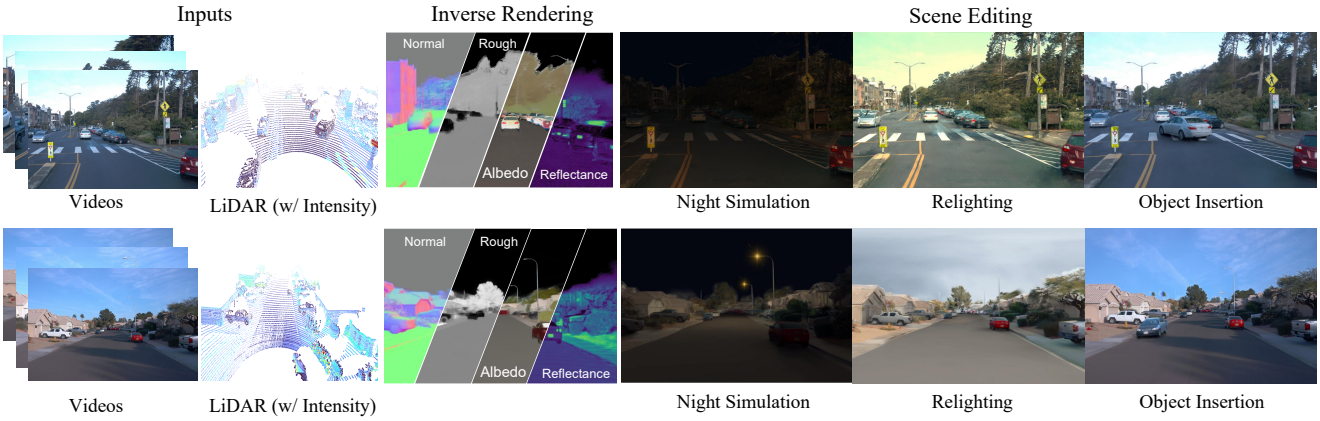

Figure 1. **Overview:** InvRGB+L takes RGB and LiDAR sequences as input and outputs a 3D scene with high-fidelity geometry, consistent albedo across RGB and LiDAR spectra, and roughness. Our representation enables photorealistic object insertion and night simulations.

Abstract

We present InvRGB+L, a novel inverse rendering model that reconstructs large, relightable, and dynamic scenes from a single RGB+LiDAR sequence. Conventional inverse graphics methods rely primarily on RGB observations and use LiDAR mainly for geometric information, often resulting in suboptimal material estimates due to visible light interference. We find that LiDAR's intensity values—captured with active illumination in a different spectral range—offer complementary cues for robust material estimation under variable lighting. Inspired by this, InvRGB+L leverages LiDAR intensity cues to overcome challenges inherent in RGB-centric inverse graphics through two key innovations: (1) a novel physics-based LiDAR shading model and (2) RGB–LiDAR material consistency losses. The model produces novel-view RGB and LiDAR renderings of urban and indoor scenes and supports relighting, night simulations, and dynamic object insertions—achieving results that surpass current state-of-the-art methods in both scene-level urban inverse rendering and LiDAR simulation.

1. Introduction

Inverse rendering is challenging because image observations are wildly ambiguous. The same image can be interpreted as a yellow wall lit by white light or as a wall that is half yellow and half white (Fig. 2); a dark region might be interpreted as a wet area or as a cast shadow. Errors like these in material recovery result in scene renderings that can be jarringly bad. Even strong material priors only partially mitigate these ambiguities (Fig. 2, middle).

LiDAR intensity provides strong cues for inverse rendering. LiDAR sensors emit laser pulses that reflect off surfaces. As is well known, time-of-flight yields geometry. We demonstrate the returned power (analogous to RGB intensity, Fig.2) provides rich surface material information. LiDAR derived material cues are extremely robust to wide changes in illumination conditions, because there is very little cross-talk between the narrow-band infrared used by LiDAR and typical illuminants indoors and outdoors. But material properties change very slowly with wavelength. So LiDAR returned power can, for example, tell that the albedo of the wall in Fig 2 is the same in the darker and lighter regions. We show that LiDAR intensity is a powerful cue that disentangles material properties and illumination effects in ways that complement SOTA methods for RGB data.

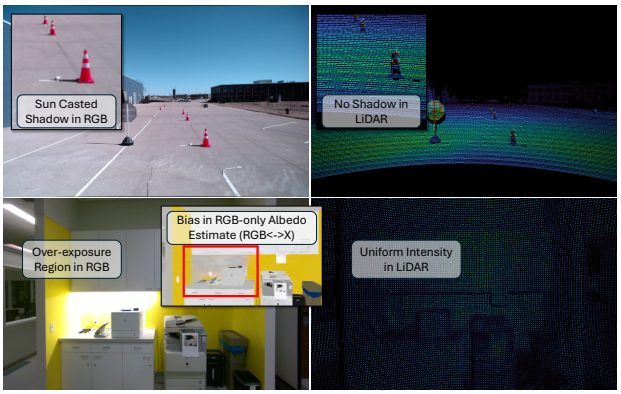

Figure 2. **Key insight:** LiDAR reflectance is less affected by environmental lighting than color images, making it an excellent complement for inverse graphics. Top: Cast shadows in color images do not appear in LiDAR reflectance; Bottom: an overexposed yellow wall shows uniform reflectance in the LiDAR spectrum.

InvRGB+L is a novel inverse rendering framework that reconstructs large, relightable, and dynamic scenes from a single RGB+LiDAR sequence. **InvRGB+L** infers geometry, illumination, and materials using *LiDAR intensity observations with color images together*, exploiting two novel technical contributions: (1) a physics-based LiDAR reflectance model that—unlike conventional reflectance models—explicitly accounts for surface specularity, and (2) a joint RGB–LiDAR material consistency loss that models the relationship between visible and LiDAR's infrared observations.

Experiments show that our model produces novel-view RGB and LiDAR renderings for both urban and indoor scenes accurately while also supporting realistic relighting, night simulations, and dynamic object insertions. Our method surpasses current state-of-the-art approaches in scene-level urban inverse rendering and novel-view LiDAR simulation in qualitative and quantitative comparisons.

2. Related Works

Inverse Rendering recovers scene properties like geometry [12, 39], materials [9, 26], and lighting [27, 38, 46] from sensor data. Light-surface interactions make the problem wildly ambiguous. Data-driven methods use dense prediction networks [3, 29, 44, 49] and diffusion models [10, 18, 21, 45] to predict intrinsic properties. The absence of an explicit physical model can result in unrealistic outcomes. Physics-based methods leverage 3D representations like NeRF [2, 4, 14, 23, 41, 48] or 3D-GS [11, 22, 30] to model geometry, then use differentiable PBR rendering to infer materials and lighting. Ambiguities remain, so priors are needed to constrain the solution space.

There exist methods that incorporate LiDAR cues [28, 41], but these do not exploit LiDAR intensity. All methods struggle with dynamic environments. In contrast, we use LiDAR intensity as a powerful cue to material properties and our method operates in dynamic environments.

LiDAR Simulation generates synthetic LiDAR data from existing observations to create new views or counterfactual scenarios. Geometry simulation has been tackled using point clouds [20], surfels [25], NeRF [13, 32–34, 42], and 3DGS [1, 5] as scene representations. Intensity simulation methods rely on lookup tables [6, 25] or encoded intensity fields [1, 13]. Many approaches mimic LiDAR ray-drop characteristics, but neglect the physics of LiDAR reflectance. In contrast, we show powerful inferences can be rooted in this physics; further, we show close attention to LiDAR physics produces better simulations. Work that models LiDAR reflection empirically [35–37, 43] assumes Lambertian surfaces. In contrast, we offer a novel formulation incorporating a specular term. Current methods produce sparse maps. In contrast we show that joint LiDAR-RGB inference results in dense, accurate maps.

3. Physics-based LiDAR Reflectance Model

LiDAR follows the rendering equation [16] and we assume no in or out scattering, so the reflected radiance is:

$$L_r(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega} f_r(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i)(\mathbf{n} \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \quad (1)$$

where $\mathbf{x}$ is the surface point, $\mathbf{n}$ is the surface normal, $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ are incident and outgoing ray directions, L_i is the incident radiance, and f_r is the BRDF at $\mathbf{x}$.

LiDAR pulses are narrow and directional, so $L_i(\mathbf{x}, \boldsymbol{\omega}_i)$ can be modelled as a constant value in a very narrow beam around $\boldsymbol{\omega}_0$ (Fig. 3 middle). Energy disperses, so the radiance at $\mathbf{x}$ will be $L_i(\mathbf{x}, \boldsymbol{\omega}_i) \propto \frac{P_e}{d^2}$, where P_e is the emitted power and d is the distance to $\mathbf{x}$. The returned beam is narrow and the sensor responds to radiance, so the sensor response is given by $I(x, \boldsymbol{\omega}_o) \propto L_r(\mathbf{x}, \boldsymbol{\omega}_o) \propto f_r(\mathbf{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_o) \frac{P_e \cos\theta}{d^2}$, where θ is the angle between $\boldsymbol{\omega}_i$ and $\mathbf{n}$.

Existing models [13, 25, 35, 37] assume Lambertian (diffuse) surfaces, where $f_r(\mathbf{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_o)$ is constant ρ_{lidar}/π, making $I \propto \frac{\rho_{\text{lidar}} P_e \cos\theta}{d^2}$, where ρ_{lidar} represents the surface reflectance (LiDAR albedo). However, this model fails to explain many real-world phenomena, such as the spotlight reflectance on metallic surfaces (e.g., cars) and water foundations.

We extend the LiDAR reflectance model by incorporating the Cook-Torrance BRDF [8], so $f_r = f_d + f_s$, where $f_d = \frac{\rho_{\text{lidar}}}{\pi}$ is the diffuse term, and f_s is the specular term. Surface roughness τ and angle θ interact, yielding $f_s \frac{F_0 \tau^2 \min(1, 2cos^2\theta)}{4\pi \cos^2\theta (\cos^2\theta(\tau^2-1)+1)^2}$, with fresnel term $F_0 = 0.04$. This specular component is a special case of the microfacet model, assuming the same incident and outgoing ray angles.

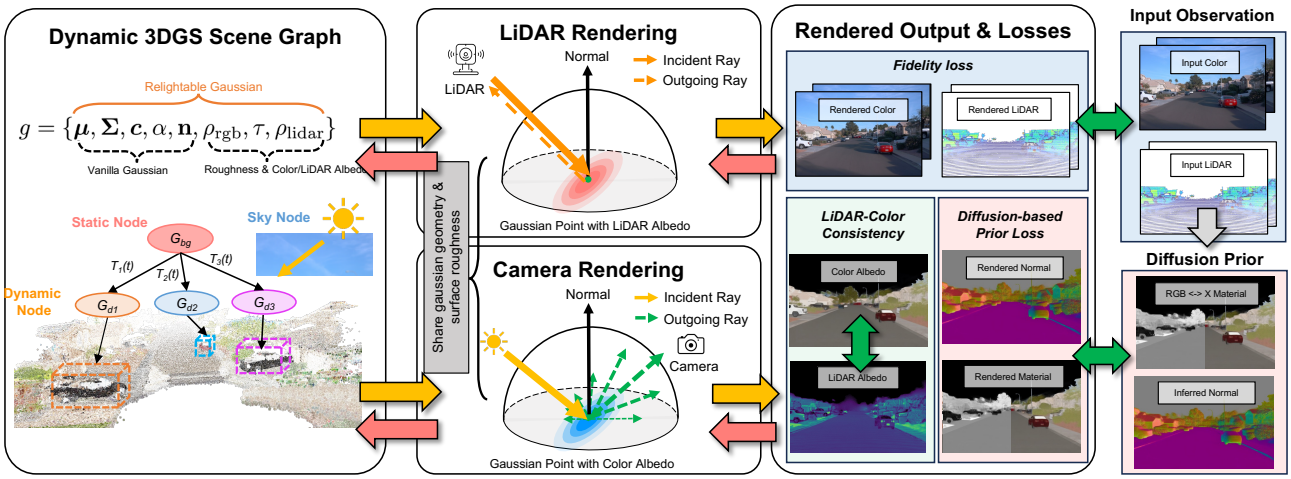

Figure 3. **Overall architecture**. We represent the scene as a dynamic, relightable 3DGS scene graph, consisting of a static node for the background, a set of dynamic nodes for movable objects, and a sky node to model illumination. Our scene can generate realistic LiDAR and camera footage via physically based forward rendering modules. Scene parameters are inferred through an inverse rendering process using backpropagation, minimizing discrepancies between rendered results and observations (as well as priors) while maximizing self-consistency. **orange arrow**: forward rendering process; **gray arrow**: diffusion-based normal and material prior inference; **red arrow**: backpropagation for inverse rendering; **green arrow**: loss computation.

Substitution yields:

$$I \propto \left(\rho_{\text{lidar}} + \frac{F_0 \tau^2 \min(1, 2cos^2\theta)}{4\cos^2\theta\left(\cos^2\theta(\tau^2-1)+1\right)^2}\right)\frac{P_e\cos\theta}{\pi d^2}. \tag{2}$$

By explicitly modeling specularity, our LiDAR reflectance model aligns with commonly used RGB-based shading models, enabling a unified framework for joint LiDAR and RGB inverse rendering in the following section. Refer to the supplementary material for details.

4. Method

We recover a **relightable 4D scene representation** that encodes geometry, color, LiDAR reflectance, and an HDR illumination model from an input video sequence $\{\mathbf{C}_t \in \mathbb{R}^{W\times H\times 3}\}_{t=0}^{T}$, LiDAR sequences $\{\mathbf{P}_t \in \mathbb{R}^{N\times 3}\}_{t=0}^{T}$ with intensity maps $\{\mathbf{I}_t \in \mathbb{R}^{W\times H\times 3}\}_{t=0}^{T}$, and their corresponding poses $\{\boldsymbol{\xi}_t \in \mathbb{SE}(3)\}_{t=0}^{T}$ captured under a single illumination environment. We represent the scene as a dynamic scene graph where each node is a 3D Gaussian encoding geometry, opacity, and intrinsic material properties for both LiDAR and camera modalities (Sec. 4.1). Forward rendering produces RGB imagery and LiDAR intensity maps from a camera pose, a scene graph and a physical model (Sec. 4.2). Inference adjusts scene parameters to produce renderings that are like observed data; our inference procedure introduce a novel albedo-consistency loss that synergizes RGB and LiDAR cues for joint reasoning (Sec. 4.3). The architecture is presented in Fig. 3.

4.1. Relightable Scene Representation

Dynamic Scene Graph We use a dynamic scene graph $\mathbf{S}$, where movable objects and backgrounds are explicitly represented as graph nodes. The representation is built out of Gaussian primitives as in 3D-GS. Each primitive $g(\boldsymbol{x})$ is defined by $g(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$, where $\boldsymbol{x} \in \mathbb{R}^3$ is a 3D coordinate, $\boldsymbol{\mu} \in \mathbb{R}^3$ stands for the mean of the Gaussian, and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3}$ is the covariance matrix. We initialize the 3D means with LiDAR points for accurate geometry. In contrast, previous work [14, 23, 41] focuses on static scenes.

The background (eg roads and buildings) is modelled with a set of static Gaussians $\mathbf{G}_{bg}$. Moving objects are represented with dynamic nodes $\{\mathbf{G}'_1, \mathbf{G}'_2, ..., \mathbf{G}'_N\}$, where N is the number of objects. Each object is represented as a 3D-GS in its local coordinate system. To place them in the dynamic scene, we apply a pose transformation $\mathbf{T}_k(t) \in \mathbb{SE}(3)$, where k is the object index and t is the timestamp. The transformed Gaussian set is then formulated as $\mathbf{G}_k(t) = \mathbf{T}_k(t) \cdot \mathbf{G}'_k$.

Illumination Model We model the sky node with spherical harmonic (SH) illumination to approximate the global lighting from the sky dome. We parameterize lighting with SH coefficients up to 3rd-order $\mathbf{L}_{\text{sky}} \in \mathbb{R}^{16\times 3}$, and define the sky lighting from incident direction ω_i as $\mathbf{L}_{\text{sky}}(\boldsymbol{\omega}_{\text{i}})$. Sky lighting fails to model sharp shadows. Additionally, we use a learnable sun light $\mathbf{L}_{sun} = \{\boldsymbol{\omega}_{sun}, I_{sun}\}$ to explicitly model directional sunlight, where $\boldsymbol{\omega}_{sun}$ is the sunlight di-

rection and I_{sun} is the sunlight intensity.

Relightable Gaussian Each of our 3D Gaussian primitives is associated with intrinsic parameters, enabling geometry and material estimation during 3D-GS optimization. We adopt the Cook-Torrance BRDF for RGB image rendering (as in the LiDAR reflectance model), so the BRDF parametrization is $f_r(\boldsymbol{\mu}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathbf{n}, \rho_{\text{rgb}}, \tau)$, where parameters are: $\mathbf{n} \in \mathbb{S}^2$ (surface normal); $\rho_{\text{rgb}} \in [0,1]^3$ (diffuse albedo); and $\tau \in [0,1]$ (surface roughness). Surface normal and surface roughness will be the same at visible and LiDAR wavelengths, but diffuse albedo may not be. We denote LiDAR albedo in the physical reflectance model as $\rho_{\text{lidar}} \in [0,1]$. For each Gaussian primitive $g(x)$, these parameters are associated to model the material properties, so: $g = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{c}, \alpha, \mathbf{n}, \rho_{\text{rgb}}, \tau, \rho_{\text{lidar}}\}$.

The entire scene representation at timestamp t is then $\mathbf{S} = \{\mathbf{G}_{\text{bg}}, \mathbf{G}_k(t), \mathbf{L}_{\text{sky}}, \mathbf{L}_{\text{sun}}\}$.

4.2. Physics-based Forward Rendering

Physics-based forward rendering of the scene serves as the foundation for inverse modeling to estimate scene parameters and supports downstream applications such as relighting and insertion rendering.

Camera Rendering We adopt a BVH-based ray tracer [11], denoted as Tracer$(\cdot)$ to trace the visibility for each Gaussian. For an incident direction $\boldsymbol{\omega}_i$, the visibility $\mathrm{v}(\boldsymbol{\omega}_\mathrm{i}) = \mathrm{Tracer}(\boldsymbol{\omega}_\mathrm{i}; \mathbf{S})$ indicates whether the Gaussian receives direct illumination from the sky. If a ray from g toward $\boldsymbol{\omega}_i$ intersects another object before reaching the sky dome, $\mathrm{v}(\boldsymbol{\omega}_\mathrm{i}) = 0$; otherwise, it is directly lit, and $\mathrm{v}(\boldsymbol{\omega}_\mathrm{i}) = 1$. However, in urban scenes, restricting ray tracing to only visible objects can lead to incomplete shadowing, as occluded objects outside the field of view may also cast shadows. To address this, we introduce an sun visibility parameter v_{sun} for each g, which indicates whether a Gaussian is directly lit by sunlight $\mathbf{L}_{\text{sun}}$ from direction $\boldsymbol{\omega}_{\text{sun}}$.

The PBR color for each Gaussian primitive can be computed using the rendering equation. We employ Monte Carlo sampling to generate M incident ray directions. Consequently, the estimated PBR color $\hat{\mathrm{c}}$ of a Gaussian primitive g for view direction ω_o is: $\hat{\mathrm{c}}(\omega_\mathrm{o}) = \frac{1}{\mathrm{M}} \sum_{\mathrm{i}=1}^{\mathrm{M}} [\mathrm{v}_{\text{sun}} \mathrm{I}_{\text{sun}}(\boldsymbol{\omega}_{\text{sun}} \cdot \mathrm{n}) + \mathrm{v}(\boldsymbol{\omega}_\mathrm{i}) \mathrm{f}_\mathrm{r}(\mu, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}; \mathbf{n}, \rho_{\text{rgb}}, \tau_{\text{rough}}) \mathbf{L}_{\text{sky}}(\boldsymbol{\omega}_\mathrm{i}) \cos\theta]$. The first term is the sunlight while the second term is the incident lighting from the sky dome. We then render the scene graph $\mathbf{S}$ into the image space through α-blending as $\hat{\mathbf{C}} = \sum_\mathrm{j} \alpha_\mathrm{j} \hat{\mathrm{c}}_\mathrm{j} \prod_{\mathrm{k}<\mathrm{j}} (1 - \alpha_\mathrm{k})$. Additionally, we render all the attributes into corresponding maps (e.g., normal map $\mathbf{N}$, albedo map $\mathbf{B}_{\text{rgb}}$, roughness map $\mathbf{R}$) using α-blending. The camera rendering results of scene graph S are defined as: $\texttt{render}_{\text{rgb}}(\mathbf{S}) = \{\hat{\mathbf{C}}, \mathbf{N}, \mathbf{B}_{\text{rgb}}, \mathbf{R}\}$.

LiDAR Rendering Given the reflectance parameter ρ_{lidar_j} and the LiDAR reflectance model in Eq. 2, we compute the intensity value for each Gaussian. Since LiDAR sensing involves a single incident ray—the laser itself—no sampling is required. The intensity I for a Gaussian is given by Eq. 2, where d and ω_o represent the distance and direction from the LiDAR origin to the Gaussian center μ, and $\cos\theta = \mathbf{n} \cdot \omega_\mathbf{o}$.We assume that the laser energy of each LiDAR channel is calibrated, setting $P_e = 1$. Finally, we render both the intensity map $\hat{\mathrm{I}}$ and the reflectance map $\mathbf{B}_{\text{lidar}}$ into image space, defining the LiDAR rendering process as: $\texttt{render}_{\text{lidar}}(\mathbf{S}) = \{\hat{\mathbf{I}}, \mathbf{B}_{\text{lidar}}\}$.

4.3. Inverse Rendering with RGB+L

Problem Formulation We must infer scene parameters – geometry, material properties, illumination, and LiDAR reflectance – from both RGB and LiDAR data. The overall loss function for optimizing the scene graph $\mathbf{S}$ is:

$$\min_{\mathbf{S}} \underbrace{\mathcal{L}_{\text{lidar}} + \mathcal{L}_{\text{rgb}}}_{\text{fidelity}} + \underbrace{\mathcal{L}_{\text{nor}} + \mathcal{L}_{\text{mat}}}_{\text{diffusion prior}} + \underbrace{\mathcal{L}_{\text{rgb}\to\text{lidar}} + \mathcal{L}_{\text{lidar}\to\text{rgb}}}_{\text{rgb}-\text{lidar consistency}} \tag{3}$$

Fidelity losses for LiDAR and RGB are:

$$\mathcal{L}_{\text{rgb}} = \|\hat{\mathbf{C}} - \mathbf{C}\|_2^2, \quad \mathcal{L}_{\text{lidar}} = \|(\hat{\mathbf{I}} - \mathbf{I}) \cdot \mathbf{M}_{\text{lidar}}\|_2^2 \tag{4}$$

where $\mathbf{C}$ is the ground-truth image and $\mathbf{M}_{lidar}$ is a mask to account for sparseness in LiDAR intensity observations. The mask is obtained by thresholding.

Diffusion-based Prior We mitigate the ambiguity in material inference by using monocular geometric and material cues from pre-trained models. We use Geowizard[10] and RGB-X[45] to preprocess multi-view training images, extracting pseudo normal and material labels, written $\hat{\mathbf{N}}$ and $\hat{\mathbf{M}} = \{\hat{\mathbf{B}}_{\text{rgb}}, \hat{\mathbf{R}}\}$ respectively. These guide inference through losses:

$$\mathcal{L}_{\text{nor}} = \|\mathbf{N} - \hat{\mathbf{N}}\|_2^2, \quad \mathcal{L}_{\text{mat}} = \|\mathbf{M} - \hat{\mathbf{M}}\|_2^2. \tag{5}$$

RGB-LiDAR Albedo Consistency Loss Spectral reflectance (which affects RGB images) and LiDAR albedo are strongly spatially correlated because each is an epiphenomenon of material microstructure (see also [19, 24]). Surfaces with similar spectral reflectance will tend to have similar LiDAR albedo and vice versa. We introduce two consistency constraints that enforce the correlation between albedo and reflectance.

LiDAR intensity maps are inherently sparse and incomplete, but spectral reflectance is a dense signal. We enforce a neighborhood smoothness constraint that propagates the sparse LiDAR albedo values into a dense map $\mathrm{B}_{\text{lidar}}$, under the assumption that it should exhibit similar smoothness as

the reflectance map $\mathbf{B}_{\text{rgb}}$. Specifically, we adopt a bilateral smoothness loss:

$$\mathcal{L}_{\text{rgb}\to\text{lidar}} = \sum_{q\in N(p)} |\mathbf{B}_{\text{lidar}_\text{p}} - \mathbf{B}_{\text{lidar}_\text{q}}| \cdot w(\mathbf{B}_{\text{rgb}_\text{p}}, \mathbf{B}_{\text{rgb}_\text{q}}), \tag{6}$$

Here, p and q denote indexes of neighboring pixels, and the weighting function is given by $w(\mathbf{B}_{\text{rgb}_\text{p}}, \mathbf{B}_{\text{rgb}_\text{q}}) = \exp\left(-\frac{(\mathbf{B}_{\text{rgb}_\text{p}} - \mathbf{B}_{\text{rgb}_\text{q}})^2}{\sigma^2}\right)$. σ is a hyperparameter controlling the sensitivity to spectral reflectance differences, ensuring smooth propagation of reflectance while preserving material boundaries.

LiDAR albedo estimates are independent of external lighting conditions, so are a powerful cue for correcting errors in reflectance estimates caused by lighting. Assume surfaces with similar albedo will have similar spectral reflectance; then we can impose a regional consistency on spectral reflectance by:

$$\mathcal{L}_{\text{lidar}\to\text{rgb}} = \sum_{\Omega} \text{var}(\mathbf{B}_{\text{rgb}_\Omega} | \mathbf{B}_{\text{lidar}_\Omega}), \tag{7}$$

where Ω is a set of regions operating as superpixels (obtained using SAM [17]) within the LiDAR albedo map $\mathbf{B}_{\text{lidar}}$. $\mathbf{B}_{rgb_\Omega}$ and $\mathbf{B}_{lidar_\Omega}$ denote the sets of spectral reflectance and albedo values within the same region Ω.

Optimization We use a two-stage optimization process. In the first stage, we supervise the scene graph without the consistency loss to obtain the initial Gaussians (only geometry, opacity and non-relightable colors) and scene graph topology. In the second stage, we fix the geometry and opacity of the Gaussians, and refine the intrinsic material properties and lighting through joint optimization. For details please refer to the supplementary materials.

5. Experiment

5.1. Experiment Protocols

Datasets We conduct experiments on the Waymo Open Dataset [31], which provides RGB images from five cameras and 64-beam LiDAR data (including intensity). uring training, we use only one camera and its corresponding LiDAR sequence. Since each scene is recorded only once, the dataset does not support quantitative relighting evaluation. To address this, we collected an additional driving scene recorded at different times of the day, capturing varying illumination conditions. We also captured an indoor scene under an artificial lighting environment to verify the effectiveness of our albedo-reflectance consistency.

Evaluation Metrics For image comparisons, we use PSNR, SSIM [40], and LPIPS [47]. For LiDAR intensity simulation, we evaluate using RMSE.

Figure 4. **Our estimated spectral reflectance vs RGB↔X.** Compared to the latest generative diffusion prior [45], our estimated spectral reflectance better reflects the vehicle's paint color and is more robust to cast shadows.

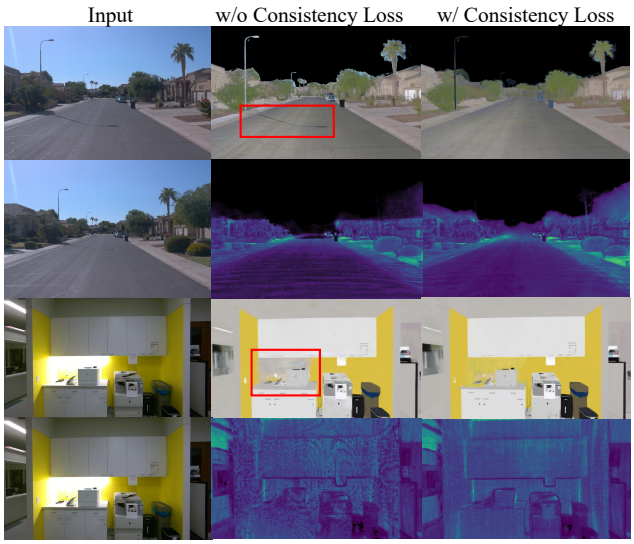

Figure 5. **The RGB-LiDAR consistency loss corrects significant errors.** Our proposed RGB-LiDAR consistency loss improves the robustness of surface reflectance estimation. In each pair of rows, **top** is spectral reflectance, **bottom** is LiDAR albedo. The cast shadow in the top pair is fixed, as is the color error around the laser printer in the second pair.

5.2. Comparison with SOTA methods

Inverse Rendering We compare our method against UrbanIR [23] and FEGR [41], two state-of-the-art approaches for urban scene inverse rendering. Since the Waymo dataset does not provide ground-truth intrinsic labels, we present only qualitative comparisons in Fig. 6, using baseline results provided by the authors of UrbanIR. Our method achieves superior inverse rendering by leveraging reflectance to effectively disentangle shading from albedo, resulting in smoother albedo estimates. In contrast, both UrbanIR and FEGR struggle to separate shadows cast by lighting poles and those beneath vehicles from the albedo, resulting in unrealistic shadows beside the car in the relight-

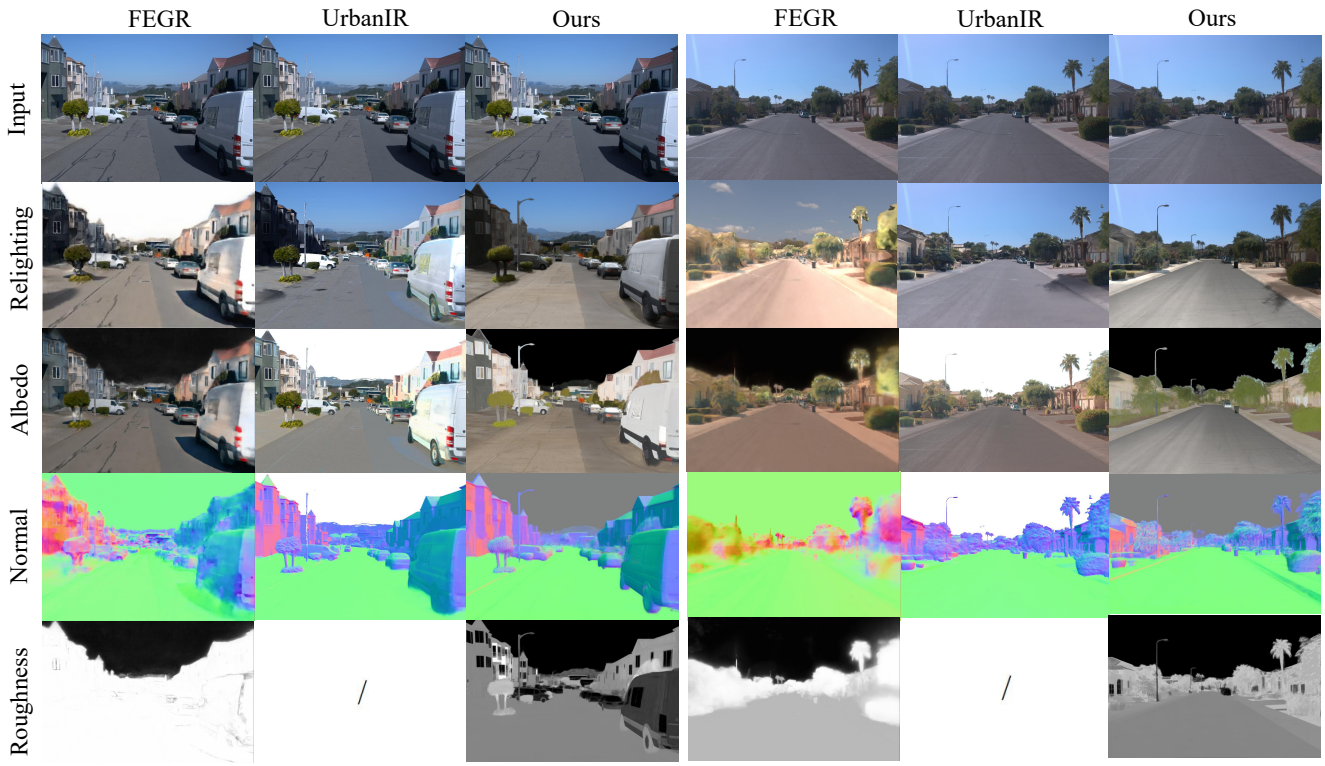

Figure 6. **Qualitative comparison for inverse rendering with FEGR and UrbanIR on Waymo dataset.** FEGR produces unrealistic normal estimates and bakes hard shadows into the albedo. UrbanIR's has no dense roughness estimation, and its radiance-based shadows cause relighting artifacts (see row 2, column 5). In contrast, our method achieves accurate and plausible material and geometry estimates, yielding superior relighting. Notice also the improved qualitative "realism" in relighting figures; surfaces tend to look more like actual object surfaces, and less like computer graphics items, likely a consequence of our roughness model. Both FEGR and our method use LiDAR, while UrbanIR relies solely on video input.

Method	PSNR↑	SSIM↑	LPIPS↓
UrbanIR [23]	28.84	0.67	0.49
w/o $\mathcal{L}_{lidar\rightarrow rgb}$	29.97	**0.73**	0.34
Ours	**30.42**	0.72	**0.30**

Table 1. **Quantitative results for relighting.**

ing results of the first scene. Additionally, we compare our albedo estimation with RGB-X [45] in Fig. 4, which serves as the diffusion-based prior for our framework. RGB-X suffers from multi-view inconsistency and fails to recover the albedo of cars. In contrast, our intrinsic 3D Gaussian representation is inherently multi-view and time-consistent, enabling it to correct erroneous predictions during training.

Relighting For quantitative evaluation, we use data captured under different lighting conditions. Specifically, we record a scene at 9 AM and 1 PM on the same day, train both sequences independently using our framework, and then replace the illumination parameters of the 9 AM scene with those from the 1 PM scene. Table 1 presents the quantitative results, where our method significantly outperforms UrbanIR. Additionally, incorporating the consistency loss further enhances performance, primarily due to more accurate material estimation. Fig. 7 shows the qualitative results, highlighting noticeable light and shadow shifts on road signs and distant buildings. In contrast, UrbanIR struggles to adjust the lighting. This demonstrates that our framework effectively disentangles illumination from albedo, enabling accurate modeling of shading variations under different lighting.

LiDAR Simulation To validate the effectiveness of our LiDAR intensity formulation and the accuracy of the generated intensity, we evaluate novel view synthesis for LiDAR intensity on the Waymo Dataset. We compare our approach against a series of LiDAR simulation works including LiDARSim [25], PCGEN [20], AlignMiF [33] and NFL [13]. Following [13], we conduct experiments on four scenes, using 50 frames from each sequence for training and selecting

Figure 7. Qualitative results for relighting. By replacing the illumination of the 9 AM scene with that of the 1 PM, we can successfully shade the tree and buildings.

Method	Intensity-RMSE ↓				
	Scene ID				Average
	1	2	3	4	
LiDARsim [25]	0.12	0.13	**0.09**	0.14	0.120
PCGEN [20]	0.11	0.15	**0.09**	0.15	0.125
AlignMiF [33]	**0.05**	**0.10**	**0.05**	0.09	**0.073**
NFL [13]	**0.06**	0.13	**0.05**	**0.08**	0.080
Ours	**0.06**	**0.08**	**0.05**	**0.06**	**0.063**

Table 2. **Quantitative results for novel view synthesis of LiDAR intensity on the Waymo Open Dataset.** The highlighted metrics denote **Best** and **Second Best**. The proposed method achieves the best results overall.

every 5th frame for validation. We use RMSE as the evaluation metric for intensity. Table 2 presents the quantitative results, where our method achieves the lowest RMSE, outperforming all baselines. This demonstrates that our formulation effectively captures the underlying physical phenomena, leading to more accurate LiDAR intensity modeling.

5.3. Ablation Studies

RGB-LiDAR Consistency Loss Fig. 5 shows qualitative comparison of inferred spectral reflectance with and without consistency loss. **Consistency removes shadows:** The shadows of the light pole are incorrectly embedded into the spectral reflectance (**first row**) when consistency is not applied. The consistency loss recovers the road correctly. **Consistency improves LiDAR albedo:** The upper part of the images in the second row is missing LiDAR albedo estimates when consistency is not applied, because the elevation range of the sensor is limited. The consistency loss propagates information from the spectral reflectance effectively propagated, filling in the missing bits. **Consistency fixes lighting induced errors:** The indoor scene of the third row shows a standard problem with estimating RGB spectral reflectance from images: spatially fast changes in illuminant baffle all intrinsic image methods, so some lighting effects get "baked" into results. The consistency loss significantly improves the accuracy of the albedo estimation.

Method	PSNR↑	SSIM↑	LPIPs↓
w/o LiDAR	33.35	0.89	0.13
w/o Dynamic	29.13	0.83	0.21
Ours	**34.76**	**0.91**	**0.11**

Table 3. **Ablation studies on rendering quality.** The highlighted metrics denote **Best**. Both LiDAR reflectance and dynamic modeling improve reconstruction quality.

Dynamic Scene Graph We conduct an ablation study to assess the effectiveness of the dynamic scene graph by removing the dynamic nodes and training a static set of Gaussians on dynamic video input. Tab. 3 presents the quantitative results for the reconstruction of PBR images on a dynamic scene in the Waymo Dataset. Besides, Fig. 8 presents the estimated albedo and roughness. As shown in the figure, without the dynamic scene graph, the moving car exhibits aliasing artifacts due to the inability to model motion. In contrast, our approach effectively captures time-varying intrinsic properties and handles the changing shadows beneath the car, enabling robust inverse rendering from dynamic video input.

LiDAR Input To evaluate the impact of LiDAR data on our framework, we conduct an ablation study where only RGB images are used. Specifically, we disable the LiDAR-based initialization of 3D-GS and exclude the albedo consistency loss term from the optimization. The reconstruction performance is reported in Table 3. While inverse rendering can still be performed without LiDAR sequence as inputs, the quality of physically based rendering (PBR) images exhibits a notable decrease. This study highlights the crucial role of LiDAR in enhancing both geometric fidelity and inverse rendering accuracy.

5.4. Downstream Applications

Scene Editing Fig. 9 presents the scene editing results, showing the versatility of our method in both relighting and object insertion. In the first row, we demonstrate night-time simulation with streetlight and headlight illumination

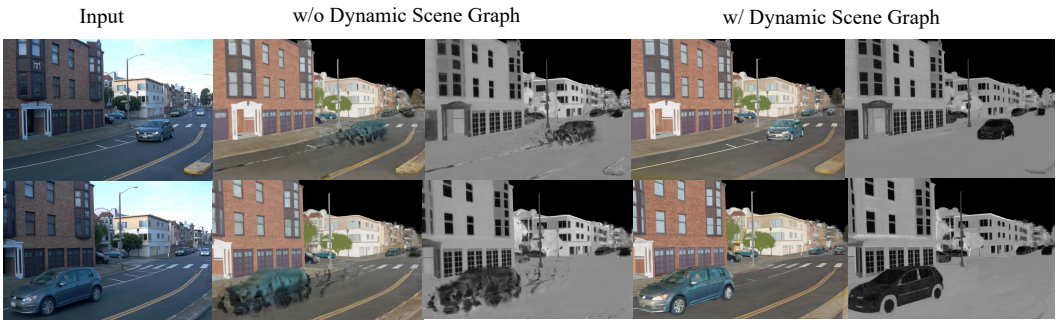

Figure 8. **Ablation study on dynamic scene graph** Explicitly modeling dynamic objects improves albedo and roughness estimation; otherwise, motion-blurred artifacts will be baked into the scene.

Figure 9. **Downstream applications of our method** Top: night simulation with controllable lights; middle: insertion with/without relighting; bottom: insertion rendering with/without changing the time of day.

to an input daytime scene. Additionally, we present object insertion results. Unlike previous approaches [23], which rely on off-the-shelf rendering engines like Blender [7], we can directly transfer a trained dynamic node from one scene to another and relight the node using our framework. The second row shows the result without relighting the inserted node: the inserted car appears mismatched with the scene. In contrast, with relighting, the car blends seamlessly into the environment. The third row shows the results of relighting both the scene and the inserted object simultaneously.

Nighttime Data Augmentation for Object Detection To evaluate the applicability of our method in autonomous driving perception, we conduct an experiment leveraging our method for nighttime data augmentation. Specifically, we transform daytime image sequences into nighttime conditions using our framework while preserving the original object detection labels. This enables the generation of nighttime training data without additional manual annotations. We generate 100 nighttime images with a total of 121 car labels which are then used to fine-tune a YOLO-v5 object detection model [15]. We evaluate the model using 50 real nighttime images from Waymo Dataset. Comparing its performance against the baseline without fine-tuning in Tab. 4, we demonstrate the potential to enhance nighttime perception for autonomous driving, particularly in low-visibility conditions, without the costly process of collecting and labeling nighttime data.

Method	Precision↑	Recall↑	mAP@50↑
w/o Night Aug.	0.537	0.281	0.236
w/ Night Aug.	**0.674**	**0.312**	**0.321**

Table 4. **Data augmentation for nighttime object detection.** Off-the-shelf object detection [15] does not perform well on Waymo nighttime sequences. Our night simulation generates nighttime logs from daytime labeled logs at no additional cost.

6. Limitation and Discussion

In this work, we integrate LiDAR into inverse rendering and introduce InvRGB+L, novel model capable of reconstructing large-scale, relightable, and dynamic scenes from a single RGB+LiDAR sequence. By leveraging the consistency between LiDAR and RGB albedo, our approach enhances material estimation and enables a variety of scene editing applications, including relighting, object insertion, and nighttime simulation. However, there are still limitations. First, we adopt a BVH-based ray tracer for 3D Gaussian ray tracing, which can produce inaccurate shadows due to the opacity properties of Gaussian primitives. Additionally, our illumination model, which accounts for only skylight and sunlight, is not sufficient for inverse rendering on complex environments such as nighttime scenes, which we will try to address in the future.

References

[1] Qifeng Chen, Sheng Yang, Sicong Du, Tao Tang, Peng Chen, and Yuchi Huo. Lidar-gs: Real-time lidar re-simulation using gaussian splatting. *arXiv preprint arXiv:2410.05111*, 2024. 2

[2] Xiaoxue Chen, Junchen Liu, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Nerrf: 3d reconstruction and view synthesis for transparent and specular objects with neural refractive-reflective fields. *arXiv preprint arXiv:2309.13039*, 2023. 2

[3] Xiaoxue Chen, Yuhang Zheng, Yupeng Zheng, Qiang Zhou, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Dpf: Learning dense prediction fields with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15347–15357, 2023. 2

[4] Xiaoxue Chen, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Time-conditioned illumination for inverse rendering of outdoor scenes. 2024. 2

[5] 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

[6] Yi-Ting Cheng, Yi-Chun Lin, and Ayman Habib. Generalized lidar intensity normalization and its positive impact on geometric and learning-based lane marking detection. *Remote Sensing*, 14(17):4393, 2022. 2

[7] Blender Online Community. Blender — a 3d modelling and rendering package, 2025. Version 4.4. 8

[8] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982. 2

[9] Timothy D Dorney, Richard G Baraniuk, and Daniel M Mittleman. Material parameter estimation with terahertz time-domain spectroscopy. *JOSA A*, 18(7):1562–1571, 2001. 2

[10] 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

[11] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024. 2, 4

[12] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 2

[13] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural lidar fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18236–18246, 2023. 2, 6, 7

[14] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023. 2, 3

[15] Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu, Adam Hogan, Laurentiu Diaconu, Jake Poznanski, Lijun Yu, Prashant Rai, Russ Ferriday, et al. ultralytics/yolov5: v3. 0. *Zenodo*, 2020. 8

[16] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*, pages 143–150. ACM, 1986. 2

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 5

[18] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for indoor single-view material estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5198–5208, 2024. 2

[19] E.L. Krinov. Spectral reflectance properties of natural formations. Technical report, National Research Council of Canada, Technical Translation: TT-439, 1947. 4

[20] Chenqi Li, Yuan Ren, and Bingbing Liu. Pcgen: Point cloud generator for lidar simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11676–11682. IEEE, 2023. 2, 6, 7

[21] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. *arXiv preprint arXiv:2501.18590*, 2025. 2

[22] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024. 2

[23] Zhi-Hao Lin, Bohan Liu, Yi-Ting Chen, Kuan-Sheng Chen, David Forsyth, Jia-Bin Huang, Anand Bhattad, and Shenlong Wang. Urbanir: Large-scale urban scene inverse rendering from a single video. *arXiv preprint arXiv:2306.09349*, 2023. 2, 3, 5, 6, 8

[24] L.T. Maloney and B.A. Wandell. A computational model of color constancy. *J. Opt. Soc. Am*, 1:29–33, 1986. 4

[25] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11167–11176, 2020. 2, 6, 7

[26] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6315–6324, 2018. 2

[27] Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Varun Jampani, Amit Raj, Pramook Khungurn, and Supasorn Suwajanakorn. Diffusionlight:

Light probes for free by painting a chrome ball. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 98–108, 2024. 2

[28] Ava Pun, Gary Sun, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, Wei-Chiu Ma, and Raquel Urtasun. Neural lighting simulation for urban scenes. *Advances in Neural Information Processing Systems*, 36:19291–19326, 2023. 2

[29] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 2

[30] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arXiv preprint arXiv:2312.05133*, 2023. 2

[31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5

[32] Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. Lidar-nerf: Novel lidar view synthesis via neural radiance fields. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 390–398, 2024. 2

[33] Tang Tao, Guangrun Wang, Yixing Lao, Peng Chen, Jie Liu, Liang Lin, Kaicheng Yu, and Xiaodan Liang. Alignmif: Geometry-aligned multimodal implicit field for lidar-camera joint synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21230–21240, 2024. 6, 7

[34] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 2

[35] Patrik Vacek, Otakar Jašek, Karel Zimmermann, and Tomáš Svoboda. Learning to predict lidar intensities. *IEEE Transactions on Intelligent Transportation Systems*, 23(4):3556–3564, 2021. 2

[36] Kasi Viswanath, Peng Jiang, PB Sujit, and Srikanth Saripalli. Off-road lidar intensity based semantic segmentation. In *International Symposium on Experimental Robotics*, pages 608–617. Springer, 2023.

[37] Kasi Viswanath, Peng Jiang, and Srikanth Saripalli. Reflectivity is all you need!: Advancing lidar semantic segmentation. *arXiv preprint arXiv:2403.13188*, 2024. 2

[38] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. Stylelight: Hdr panorama generation for lighting estimation and editing. In *European conference on computer vision*, pages 477–492. Springer, 2022. 2

[39] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[41] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023. 2, 3, 5

[42] Hanfeng Wu, Xingxing Zuo, Stefan Leutenegger, Or Litany, Konrad Schindler, and Shengyu Huang. Dynamic lidar resimulation using compositional neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19988–19998, 2024. 2

[43] Qiong Wu, Ruofei Zhong, Pinliang Dong, You Mo, and Yunxiang Jin. Airborne lidar intensity correction based on a new method for incidence angle correction for improving land-cover classification. *Remote Sensing*, 13(3):511, 2021. 2

[44] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 2

[45] 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, 4, 5, 6

[46] Fangneng Zhan, Changgong Zhang, Yingchen Yu, Yuan Chang, Shijian Lu, Feiying Ma, and Xuansong Xie. Emlight: Lighting estimation via spherical distribution approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3287–3295, 2021. 2

[47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 5

[48] Jingsen Zhu, Yuchi Huo, Qi Ye, Fujun Luan, Jifan Li, Dianbing Xi, Lisha Wang, Rui Tang, Wei Hua, Hujun Bao, et al. I2-sdf: Intrinsic indoor scene reconstruction and editing via raytracing in neural sdfs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12489–12498, 2023. 2

[49] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2822–2831, 2022. 2