SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Neural Scene Decomposition for Accurate Light and Material Reconstruction via Physically-Based Global Illumination Estimation

Yue Chen

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Neural Scene Decomposition for Accurate Light and Material Reconstruction via Physically-Based Global Illumination Estimation

Neuronale Szenenzerlegung für präzise Licht- und Materialrekonstruktion durch physikalisch-basierte globale Beleuchtungsschätzung

Author:	Yue Chen
Supervisor:	Matthias Niessner
Advisor:	Peter Kocsis
Submission Date:	16.10.2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 16.10.2023 Yue Chen

Acknowledgments

I would like to express my gratitude to my advisor, Peter Kocsis, for his guidance and support. I am grateful to my supervisor Prof. Matthias Niessner for introducing me to the world of neural networks. I want to thank Christoph Weiler for his remote assistance. I appreciate my teammates, Mohamed Ebbed and Burak Çuhadar, with whom this thesis idea was born. Grateful thanks to my girlfriend, Minghua Chen, and my family. Special thanks to those who generously provided tutorials and insights into physically-based rendering, including the PBR book authors, the Mitsuba team, and Prof. Cem Yuksel.

Abstract

Recent advances in neural rendering have achieved pinpoint reconstruction of 3D scenes from multi-view images. To enable scene editing under different lighting conditions, an increasing number of methods are integrating differentiable surface rendering into the pipelines. However, many of these methods rely heavily on simplified surface rendering algorithms, while considering primarily direct lighting or fixed indirect illumination only. We introduce a more realistic rendering pipeline that embraces multi-bounce Monte Carlo (MC) path tracing. Benefiting from the multi-bounce light path estimation, our method can decompose high-quality material properties without necessitating additional prior knowledge. Additionally, our model can accurately estimate and reconstruct secondary shading effects, such as indirect illumination and self-reflection. We demonstrate the advantages of our model to baseline methods qualitatively and quantitatively across synthetic and real-world scenes.

Contents

1 Introduction

The reconstruction of a scene and the synthesis of its controllable and photo-realistic images based on multi-view images is a longstanding challenge in both computer vision and computer graphics [Tew+21]. This reconstruction facilitates various applications, including novel view synthesis, relighting, and material editing. However, this task is challenging and ill-posed due to ambiguities in the potential output solutions. NeRF [Mil+20] and succeeding methods [Bar+22; Fri+22] employ differentiable volume rendering to achieve high-quality view interpolation via density-based light fields. They excel in novel view synthesis but struggle in material editing and relighting tasks.

Surface-based methodologies utilize differentiable surface rendering to disentangle the scene into its geometry, material properties, and lighting, thereby enabling precise scene editing. However, many of these models rely heavily on simplified surface rendering algorithms, while considering primarily direct lighting [Zha+21a; Zha+21b; Mun+22; HHM22] or fixed indirect illumination [Zha+22; Jin+23]. They often struggle in the reconstruction of secondary shading effects like self-reflection.

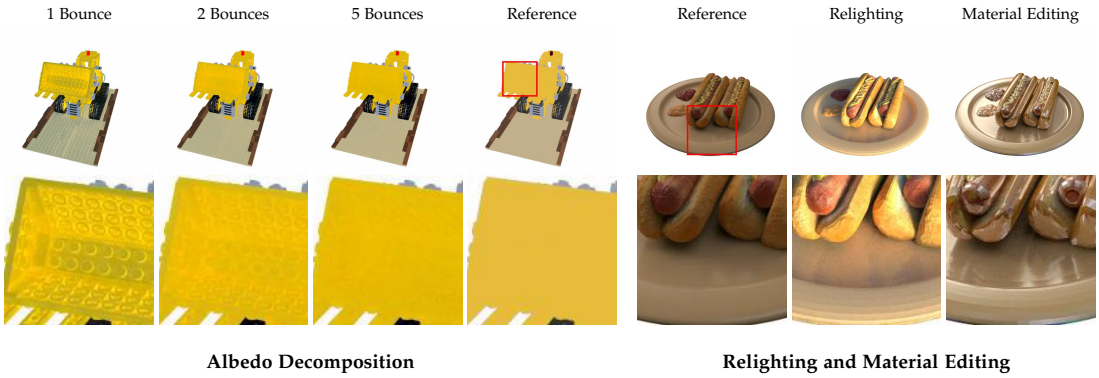

Figure 1.1: **Visualization of the advantages of multi-bounce global illumination estimation.** Our model adeptly decomposes material properties, reducing inaccuracies where lighting gets embedded into the material, all without relying on additional prior knowledge. Moreover, it effectively reconstructs secondary shading effects, like self-reflection, leading to superior relighting and material editing outcomes.

In this work, we introduce a physically-based neural rendering pipeline that integrates differentiable multi-bounce MC path tracing. Our model can jointly estimate scene materials and illumination from multi-view images taken under a singular, unknown lighting condition. As illustrated in Figure 1.1, the simulation of multi-bounce light transport enables a precise reconstruction of the material properties, which is distinct from preceding models that heavily rely on prior knowledge about the material (such as smoothness for material parameters [Zha+21b; HHM22; Jin+23]). Moreover, our model can effectively capture and reproduce secondary shading effects like color bleeding and self-reflection. This capability leads to a more precise decomposition of material properties and an enhanced relighting performance.

In summary,

- We propose a differentiable multi-bounce MC path tracer that achieves scene decomposition into Physically Based Rendering (PBR) materials and lighting.
- Our model includes physically-based global illumination estimation, facilitating the precise reconstruction of secondary shading effects such as self-reflection.
- Our model can reconstruct high-quality material properties, eliminating the need for additional prior knowledge.

2 Related Work

Neural rendering for multi-view reconstruction Neural rendering approaches can be broadly categorized into two main types: volume rendering and surface rendering. NeRF [Mil+20] and NeRF-like methods [Fri+22; Bar+22; Bar+23] leverage the implicit absorption-emission volumetric rendering model, treating the scene as a volume of particles that absorb and emit light. While exceeding the task of novel view synthesis, those models are incapable of tasks like relighting and material editing. On the other hand, surface rendering techniques are adept at simulating light transport and analyzing an object surface scattering properties. Some models adopt implicit surface representations, either through a Signed Distance Function (SDF) [Zha+21a; Zha+22] or a volumetric density field [Zha+21b; Jin+23; Mai+23]. Others [Mun+22; HHM22] back-propagate directly to explicit triangle meshes. In our work, we employ physically-based MC path tracing, an accurate, unbiased surface rendering technique.

Material and illumination modeling The majority of surface-based methodologies leverage the PBR material model [Zha+21a; Bos+21; Sri+21; Zha+22; HHM22; Jin+23; Mai+23] and predict its spatially varying parameters. Some methods such as [Zha+21b] employ neural networks to predict the Bidirectional Reflectance Distribution Function (BRDF). For representing illumination, techniques range from using mixtures of spherical Gaussians [Zha+21a; Zha+22; Jin+23] to HDR environment maps [Zha+21b; Mun+22; HHM22; Mai+23]. In our work, we employ a spatially varying PBR microfacet material model, while its parameters are determined by a neural field. To capture high-frequency details, we utilize an HDR environment map for illumination.

Indirect Illumination Estimation For volume-based methodologies, the indirect illumination is intrinsically embedded into the radiance fields. However, they are not editable. Surface-based methods often struggle to estimate indirect illumination due to the high computational cost. Most surface rendering methods [Zha+21a; Bos+21; Sri+21; Zha+21b; HHM22] ignore indirect illumination. As a result, secondary shading effects are baked into the material ground color, which leads to noisy outputs. They heavily rely on additional prior knowledge to prevent this issue. Other models [Zha+22; Jin+23] represent indirect illumination as a radiance field. While they can reconstruct the indirect illumination for the training scene, they can not reconstruct realistic secondary

shading effects under new lighting conditions. Our model addresses this issue by leveraging path tracing, which can perform online multi-bounce calculations at an affordable cost.

3 Background

3.1 Scene Representations

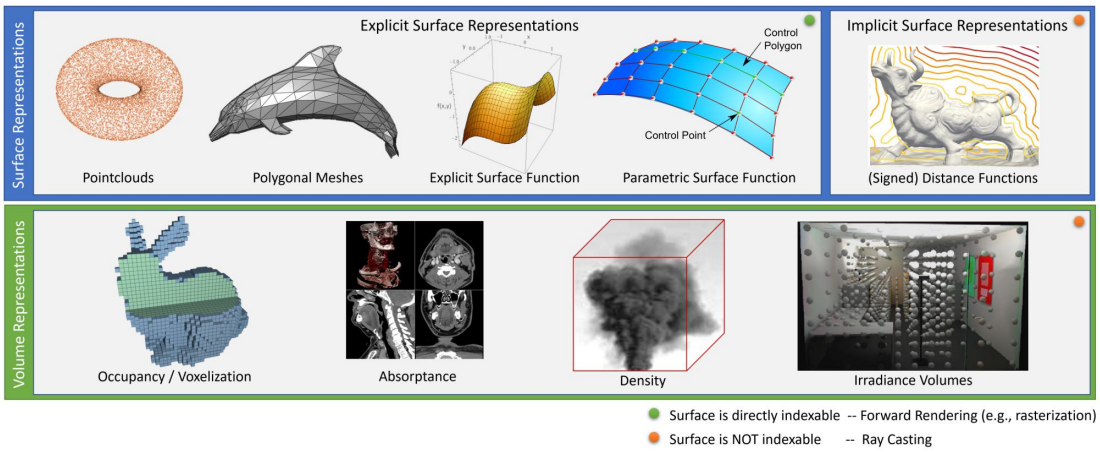

Figure 3.1: **An overview of classical surface and volume representations.** Image from [Tew+21]

The surface representations can be categorized into surface representations and volume representations, as shown in Figure 3.1. Volume representations can capture volumetric properties like densities and occupancies. In contrast, surface representations represent specific properties related to an object's surface. In general, volumetric representations can represent surfaces, but not vice versa [Tew+21]. For instance, model volumetric matter like smoke and fog can not be represented using surface representations. In neural rendering, both volumetric and surface representations are frequently employed. Each offers distinct advantages and disadvantages.

3.2 The Rendering Equation

The Rendering Equation [Kaj86] is the foundation of PBR. It describes the total amount of light emitted from a point on object along a particular viewing direction. The

equation is expressed as:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| \, d\omega_i, \tag{3.1}$$

where $L_o(p, \omega_o)$ represents the the outgoing radiance viewing from a point p in direction ω_o. $L_e(p, \omega_o)$ is the emitted radiance at a point p towards direction ω_o. $L_i(p, \omega_i)$ stands for the incident radience from direction ω_i, $f(p, \omega_o, \omega_i)$ for the BRDF and $|\cos \theta_i|$ for the geometry term.

The evaluation of the rendering equation is recursive. The calculation of an analytical solution is generally unfeasible. One must either make simplified assumptions, such as estimating direct illumination only, or use to numerical integration methods like the MC path tracer. Figure 3.2 shows the difference between using direct illumination and using MC path tracer.

Estimating direct illumination only

Using MC path tracer for global illumination

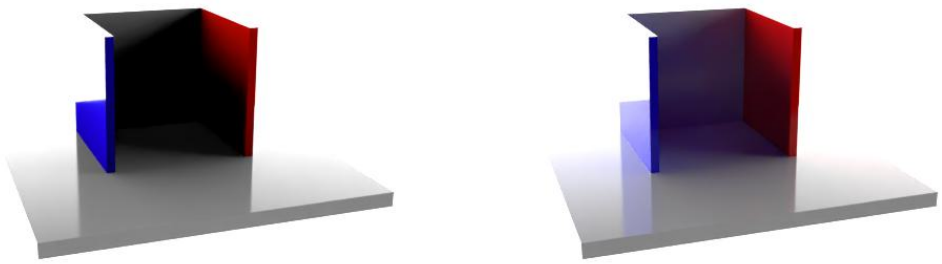

Figure 3.2: **Comparison of rendering with vs. without Global Illumination.** The global illumination is crucial for rendering realistic images. The image rendered with MC path tracer contains secondary effects, for instance, self-reflection and a brighter shadow.

3.3 Reflection Models

To describe how the surface scatters light, BRDF and Bidirectional Transmittance Distribution Function (BTDF) are defined. While BRDF defines how light is reflected at the surface, BTDF measures the behavior of light transmittance. Bidirectional Scattering Distribution Function (BSDF) is often a mixture of multiple BRDF and BTDF for modeling the scattering of real-world material. Figure 3.3 visualized renderings using BRDF and BTDF. In our work, we model the scene material properties using BRDF because we only take reflection into account.

The simplest BRDF model is Lambertian's diffuse model:

$$f = \frac{R_{\mathrm{d}}}{\pi}, \tag{3.2}$$

where R_{d} stands for Lambertian reflectance. It models the behavior of light scattered at perfectly diffuse surfaces.

For a perfectly smooth surface, its BRDF has the formula:

$$f = F_{\mathrm{r}}\left(\omega_{\mathrm{r}}\right) \frac{\delta\left(\omega_{\mathrm{i}} - \omega_{\mathrm{r}}\right)}{\left|\cos\theta_{\mathrm{r}}\right|}, \tag{3.3}$$

where F_{r} is the Fresnel term, and δ is the delta distribution.

The Phong reflection model [Pho98] is introduced to simulate the specular reflectance:

$$f = k_s \frac{n+2}{2\pi} \cos^n \alpha, \tag{3.4}$$

while k_s represents the specular reflectivity, and n represents the specular exponent. α is the angle between the perfect specular reflective direction and the outgoing ray direction.

The Torrance–Sparrow Microfacet model [TS67] offers a more expressive representation for capturing the reflection interactions of light. It has the formula:

$$f = \frac{D\left(\omega_{\mathrm{h}}\right) G\left(\omega_{\mathrm{o}}, \omega_{\mathrm{i}}\right) F_{\mathrm{r}}\left(\omega_{\mathrm{o}}\right)}{4\cos\theta_{\mathrm{o}}\cos\theta_{\mathrm{i}}}, \tag{3.5}$$

where D stands for a microfacet normal distribution function, G for the a masking-shadowing function, and F_{r} for Fresnel term. In general, two types of microfacet distribution are often used, the Beckmann–Spizzichino [BS87] and Trowbridge–Reitz (GGX) [TR75].

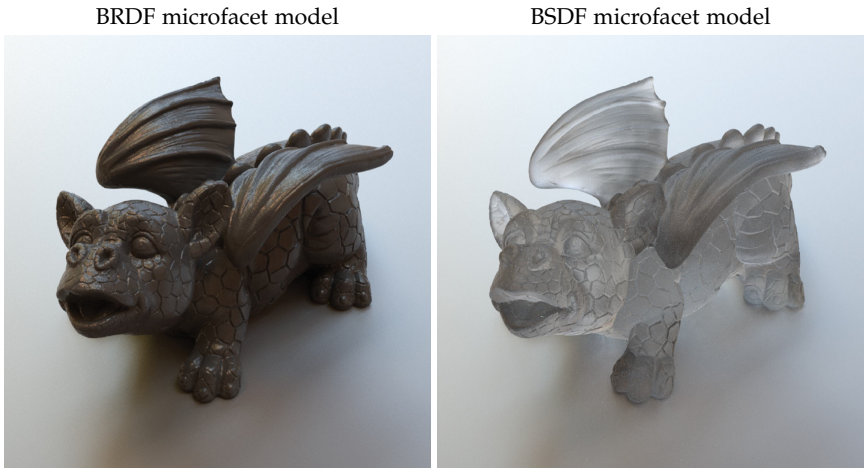

Figure 3.3: **Comparison of rendering using BRDF and BTDF.** BRDF can model reflection, while BTDF models transmission. Image from [PJH16]

3.4 Monte Carlo

MC can serve as a universal approximation of an arbitrary integral. Given an integral $\int_a^b f(x)\mathrm{d}x$, it can be approximated by:

$$F_N = \frac{b-a}{N}\sum_{i=1}^{N} f\left(X_i\right), \tag{3.6}$$

where $X_i \in [a, b]$ is a random uniform random variables.

The restriction to uniform random variables can be relaxed:

$$F_N = \frac{1}{N}\sum_{i=1}^{N} \frac{f\left(X_i\right)}{p\left(X_i\right)}, \tag{3.7}$$

as long as X_i are drawn from a Probability Density Function (PDF) $p\left(X_i\right)$.

In general, when $p\left(X_i\right) \propto f(x)$, F_N becomes more robust against variance. The choice of $p\left(X_i\right)$ is a crucial point for the accuracy of the approximation. In PBR, we often use sampling strategies like BRDF sampling, emitter sampling, and Multiple Importance Sampling (MIS).

The Rendering Equation can be approximated by MC:

$$\begin{aligned} L_\mathrm{o}\left(\mathrm{p}, \omega_\mathrm{o}\right) &= \int_{\mathrm{S}^2} f\left(\mathrm{p}, \omega_\mathrm{o}, \omega_\mathrm{i}\right) L_\mathrm{i}\left(\mathrm{p}, \omega_\mathrm{i}\right) \left|\cos\theta_\mathrm{i}\right| \mathrm{d}\omega_\mathrm{i} \\ &\approx \frac{1}{N}\sum_{j=1}^{N} \frac{f\left(\mathrm{p}, \omega_\mathrm{o}, \omega_j\right) L_\mathrm{i}\left(\mathrm{p}, \omega_j\right) \left|\cos\theta_j\right|}{p\left(\omega_j\right)}. \end{aligned} \tag{3.8}$$

3.5 Multiple Importance Sampling

MIS [VG95] is a powerful algorithm that combines the advantage of two sampling strategies. To estimate the value of $\int_a^b f(x)g(x)\mathrm{d}x$, two sampling distributions p_f and p_g are available. We can sample both distributions and combine their strengths by:

$$\frac{1}{n_f}\sum_{i=1}^{n_f}\frac{f\left(X_i\right)g\left(X_i\right)w_f\left(X_i\right)}{p_f\left(X_i\right)}+\frac{1}{n_g}\sum_{j=1}^{n_g}\frac{f\left(Y_j\right)g\left(Y_j\right)w_g\left(Y_j\right)}{p_g\left(Y_j\right)}, \tag{3.9}$$

with the power heuristic (i.e. MIS weight):

$$w_s(x)=\frac{\left(n_s p_s(x)\right)^\beta}{\sum_i\left(n_i p_i(x)\right)^\beta}. \tag{3.10}$$

In practice, β is typically set to 2. Figure 3.4 illustrates how MIS can combine two sampling methods while retaining their respective strengths.

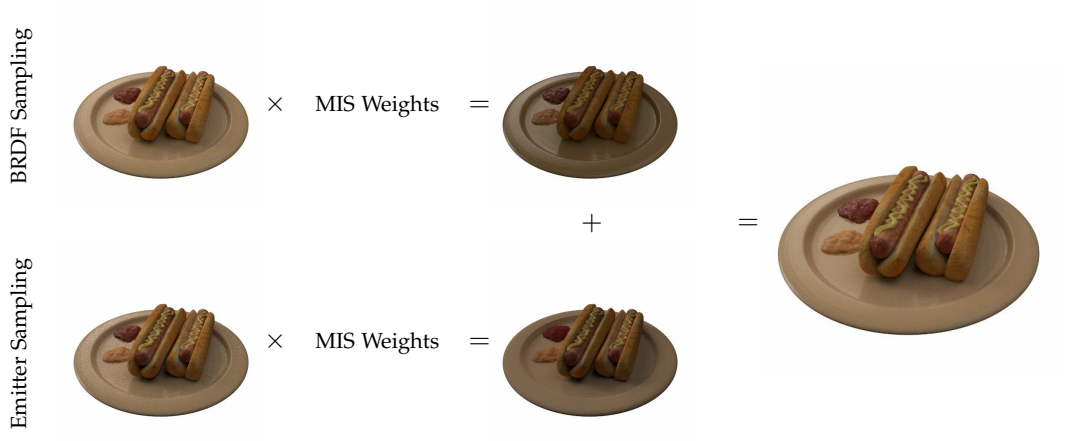

Figure 3.4: **Visualization of the contribution of MIS.** MIS leverages the strengths of BRDF sampling and emitter sampling, resulting in less noise rendering.

4 Methodology

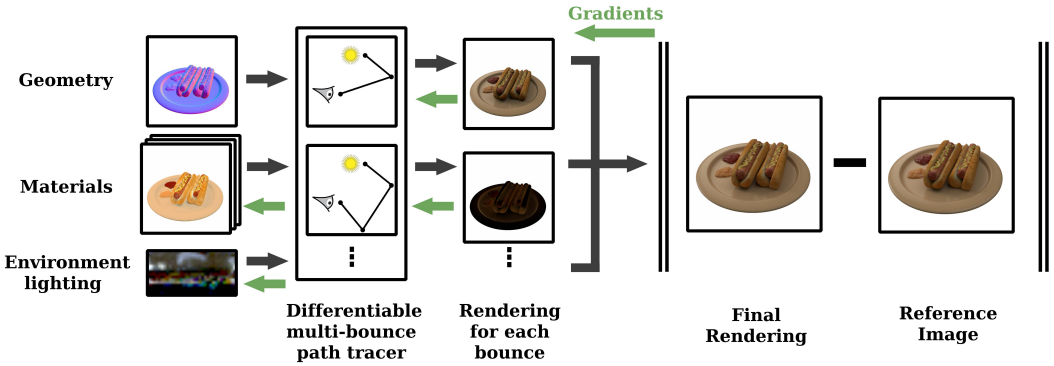

Figure 4.1: **Overview of the pipeline.** We present a differentiable multi-bounce MC path tracer, designed for an online rendering of images at each light path bounce. The system uses a pre-optimized mesh geometry, spatially varying materials, and environment lighting as input. We supervise the training solely through photometric loss, with gradients being back-propagated to the path tracer, materials, and environment lighting.

In this chapter, we introduce our neural scene decomposition model, as depicted in Figure 4.1. Our primary objective is to leverage multi-bounce physically-based rendering to reconstruct accurate lighting and material properties of a scene.

To focus on the rendering pipeline and assess the feasibility of multi-bonce optimization, we adopt a pre-optimized mesh representation. We apply Mitsuba3 [Jak+22] to perform ray tracing, which efficiently determines surface intersections. For BRDF parameter optimization, we utilize Tiny CUDA Neural Networks [Mül21]. Additionally, a fixed-size tensor is employed to store environmental lighting information. During the rendering phase, we introduce a differentiable path-tracing algorithm that permits multiple bounces to render images. Subsequently, an L2 loss is computed between the predicted images and the ground truth. The gradient is then back-propagated through the renderer to both the material networks and the lighting tensor.

The subsequent sections first outline the underlying assumptions of our model. Thereafter, we dive into the specifics of each model component: the geometry, material, lighting, and rendering algorithm.

4.1 Assumption

Firstly, we assume that the object in question has hard surfaces. In doing so we do not take volumetric matters like smoke and fog into consideration, which allows us to employ an explicit mesh surface representation.

Secondly, we assume that the scene consists solely of isotropic reflective materials, excluding considerations of transmission or polarization. This assumption permits the use of a BRDF for modeling light scattering on the geometry surface.

Lastly, we assume the scene is illuminated solely by an ambient light source located at a considerable distance from the rendering object. The object itself does not include emitters such as light sources or fluorescent materials. This assumption facilitates the usage of a latitude–longitude radiance map to model the light of the scene.

4.2 Geometry and Ray Casting

The primary goal of our model is to explore the potential and performance of multi-bounce global illumination estimation. To this end, we utilize a fixed mesh representation of the scene. Specifically, we adopt the pre-optimized mesh from TensoIR [Jin+23] for synthetic scenes, and MonoSDF [Yu+22] for real-world scenes. To delve deeper into the impact of multi-bounce light path estimation, we also incorporate Ground Truth (GT) mesh derived from synthetic datasets. We employ Mitsuba3 [Jak+22] for ray tracing in order to efficiently identify the intersections between the rays and the object.

We first need to derive a mesh geometry. We use the marching cube algorithm [LC87] to extract mesh from an implicit surface representation of a pre-optimized model. Notably, the resulting mesh often bears numerous artifacts, as shown in Figure 4.2. This is because the marching cube method discretizes the object into small surfaces based on a set resolution. Such discretization compromises the precision of the surface.

It is also worth mentioning that the multi-bounce global illumination estimation does not mandate the usage of mesh representation and ray tracing. The key requirement is the ability to accurately identify surface intersections and the corresponding normals.

4.3 Material and Lighting

For representing diffuse reflection, we use the Lambertian reflection model:

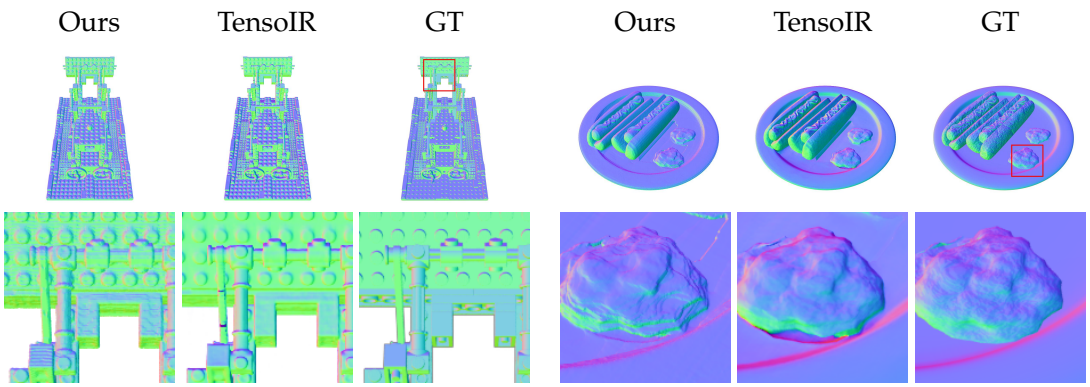

Figure 4.2: **Visulization of the noisy surfaces generated by marching cube algorithms.** Our model employs a mesh geometry representation, which is derived from a pre-optimized implicit geometry representation using marching cube algorithms. The surfaces of the extracted mesh often bear numerous artifacts.

$$f_{\text{diffuse}} = \frac{R_{\text{d}}}{\pi}, \tag{4.1}$$

where R_{d} stands for Lambertian reflectance.

For specular reflection, we employ the Torrance–Sparrow model [TS67]:

$$f_{\text{specular}}\left(\omega_{\text{o}}, \omega_{\text{i}}\right) = \frac{D\left(\omega_{\text{h}}\right) G\left(\omega_{\text{o}}, \omega_{\text{i}}\right) F_{\text{r}}\left(\omega_{\text{o}}\right)}{4 \cos\theta_{\text{o}} \cos\theta_{\text{i}}}, \tag{4.2}$$

while ω_{o}, ω_{i}, and ω_{h} are the solid angle of outgoing direction, incident direction, and their half angle respectively. $D\left(\omega_{\text{h}}\right)$ stands for the microfacet normal distribution function, $G\left(\omega_{\text{o}}, \omega_{\text{i}}\right)$ for the masking-shadowing function. We adopt the Trowbridge–Reitz microfacet distribution (GGX) [TR75]. For the Fresnel function $F_{\text{r}}\left(\omega_{\text{o}}\right)$, we follow the approximation from Schlick [Sch93]:

$$F_{\text{r}}(\cos\theta) = R_{\text{s}} + (1 - R_{\text{s}})(1 - \cos\theta)^5, \tag{4.3}$$

in which R_{s} stands for the specular reflectance of the surface at normal incidence.

These components are then aggregated to form our comprehensive BRDF model:

$$f_{\text{r}}\left(\omega_{\text{o}}, \omega_{\text{i}}\right) = f_{\text{diffuse}} + f_{\text{specular}}\left(\omega_{\text{o}}, \omega_{\text{i}}\right). \tag{4.4}$$

We employ Tiny CUDA Neural Networks [Mül21], which contains a fully fused multilayer perception and a versatile multiresolution hash positional encoding, to predict

material properties $\mathbf{m}$. For a given surface intersection x_{surf}, its material properties $\mathbf{m}$ is equal to $(R_{\text{d}}, R_{\text{s}}, \textit{roughness})$, where *roughness* represents the roughness of the surface, which is a parameter for microfacet function $D(\omega_{\text{h}})$ and $G(\omega_{\text{o}}, \omega_{\text{i}})$. The mapping can be written as:

$$MLP : x_{\text{surf}} \mapsto \mathbf{m}. \tag{4.5}$$

We utilize a straightforward lighting representation: an HDR latitude-longitude environment lighting image, i.e. an infinitely distant area light source enveloping the entire scene. This representation enables detailed high-frequency lighting.

4.4 Differentiable Multi-bounce Monte Carlo Path Tracer

The Rendering Equation [Kaj86] can be estimated with MC [PJH16]:

$$\begin{aligned} L_{\text{o}}(\text{p}, \omega_{\text{o}}) &= \int_{\text{S}^2} f(\text{p}, \omega_{\text{o}}, \omega_{\text{i}}) \, L_{\text{i}}(\text{p}, \omega_{\text{i}}) \, |\cos\theta_{\text{i}}| \, \text{d}\omega_{\text{i}} \\ &\approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(\text{p}, \omega_{\text{o}}, \omega_j) \, L_{\text{i}}(\text{p}, \omega_j) \, |\cos\theta_j|}{p(\omega_j)}, \end{aligned} \tag{4.6}$$

with directions ω_{j} sampled from a distribution with respect to solid angle that has PDF $p(\omega_j)$. $f(\text{p}, \omega_{\text{o}}, \omega_j)$ denotes the BRDF. $L_{\text{o}}(\text{p}, \omega_{\text{o}})$ represents the outgoing radiance at surface point p viewed from ω_o, $L_{\text{i}}(\text{p}, \omega_j)$ represents the incident radiance arriving at p along ω_j.

In the context of a multi-bounce light path setup, determining the value of $L_{\text{i}}(\text{p}, \omega_j)$ becomes recursive. The reason for this is that the incident radiance encapsulates the outgoing radiance from earlier bounces. As a result, the computational effort is greatly complicated.

We incorporate path tracing to tackle this issue. In path tracing, instead of evaluating the full rendering equation for each secondary ray, we trace a single light path and calculate its outgoing radiance as it bounces around the scene multiple times. The outgoing radiance for one light path is the sum of the radiance contribution of each bounce:

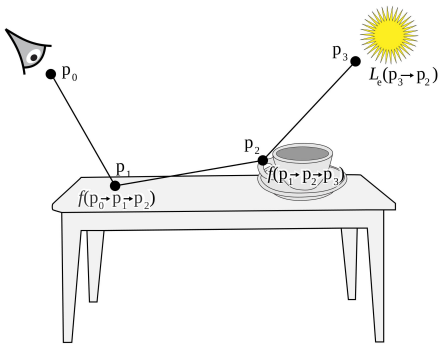

Figure 4.3: **The three-point form of the light transport equation.** This form effectively represents the directions of light paths and the corresponding bounce count. Note that $f\left(\mathrm{p}_{l-1} \rightarrow \mathrm{p}_l \rightarrow \mathrm{p}_{l+1}\right)$ is equivalent to $f\left(\mathrm{p}_{l+1} \rightarrow \mathrm{p}_l \rightarrow \mathrm{p}_{l-1}\right)$ due to the reciprocity property of the BRDF. The image is modified from PBR book [PJH16].

$$
\begin{aligned}
\frac{f\left(\mathrm{p}, \omega_{\mathrm{o}}, \omega_j\right) L_{\mathrm{i}}\left(\mathrm{p}, \omega_j\right)\left|\cos \theta_j\right|}{p\left(\omega_j\right)} =& \frac{f\left(\mathrm{p}_2 \rightarrow \mathrm{p}_1 \rightarrow \mathrm{p}_0\right)\left|\cos \theta_{\mathrm{p}_2 \leftrightarrow \mathrm{p}_1}\right|}{p\left(\mathrm{p}_2 \leftrightarrow \mathrm{p}_1\right)} * L_{\mathrm{e}}\left(\mathrm{p}_2 \rightarrow \mathrm{p}_1\right) \\
&+ \frac{f\left(\mathrm{p}_2 \rightarrow \mathrm{p}_1 \rightarrow \mathrm{p}_0\right)\left|\cos \theta_{\mathrm{p}_2 \leftrightarrow \mathrm{p}_1}\right|}{p\left(\mathrm{p}_2 \leftrightarrow \mathrm{p}_1\right)} * \\
& \frac{f\left(\mathrm{p}_3 \rightarrow \mathrm{p}_2 \rightarrow \mathrm{p}_1\right)\left|\cos \theta_{\mathrm{p}_3 \leftrightarrow \mathrm{p}_2}\right|}{p\left(\mathrm{p}_3 \leftrightarrow \mathrm{p}_2\right)} * L_{\mathrm{e}}\left(\mathrm{p}_3 \rightarrow \mathrm{p}_2\right) \\
&+ \ldots \\
=& \sum_{k=1}^{k_{max}} \beta_k * L_{\mathrm{e}}\left(\mathrm{p}_{k+1} \rightarrow \mathrm{p}_k\right),
\end{aligned}
\tag{4.7}
$$

$$
\text{with } \beta_{\mathrm{k}} = \prod_{l=1}^{k} \frac{f\left(\mathrm{p}_{l+1} \rightarrow \mathrm{p}_l \rightarrow \mathrm{p}_{l-1}\right)\left|\cos \theta_{\mathrm{p}_{l+1} \leftrightarrow \mathrm{p}_l}\right|}{p\left(\mathrm{p}_{l+1} \leftrightarrow \mathrm{p}_l\right)}.
$$

We adopt the three-point form (Figure 4.3) of the light transport equation. k represents the current bounce count, while k_{max} is the maximal permissible number of bounce. $L_{\mathrm{e}}\left(\mathrm{p}_{k+1} \rightarrow \mathrm{p}_k\right)$ is the emitter radiance shooting from surface point p_{k+1} to p_k. Note that if p_{k+1} is located on the surface of an emitter, then the emitter radiance term is non-zero. Additionally, $L_{\mathrm{e}}\left(\mathrm{p}_{k+1} \rightarrow \mathrm{p}_k\right)$ includes a visibility test between p_k and p_{k+1}, which is omitted in the equation above. $f\left(\mathrm{p}_{l+1} \rightarrow \mathrm{p}_l \rightarrow \mathrm{p}_{l-1}\right)$ represents the BRDF,

characterizing the reflection properties of the material at surface point p_l, where the outgoing direction being from p_l to p_{l-1} and the incident direction from p_l to p_{l+1}.

Furthermore, we utilize MIS [VG95], a strategy that assigns weights to different sampling methods to minimize noise in MC integration. This process is illustrated in Figure 3.4. More precisely, we employ MIS using two sampling techniques: BRDF sampling, and emitter sampling. After each optimization iteration, we recalculate the sampling distribution based on the latest scene parameters.

Following the taxonomy of differentiable MC estimators of [Zel+21], our model adopts a *detached* sampling strategy. Instead of calculating the complete gradients of the scene parameters, we approximate only a partial of the gradient, similar as in NVDIFFRECMC [HHM22]. In more detail, the gradient of the integration of function $I = g(\mathbf{m})$ with respect to scene parameter $\mathbf{m}$ is calculated through:

$$\partial_{\mathbf{m}} \int I dx = \partial_{\mathbf{m}} \int g(\mathbf{m}) dx. \tag{4.8}$$

First we apply the MC estimation, where p_i represents the PDF of the i-th sample:

$$\partial_{\mathbf{m}} \int I dx \approx \partial_{\mathbf{m}} (\frac{1}{N} \sum_{i}^{N} \frac{g(\mathbf{m})}{p_i(\mathbf{m})}). \tag{4.9}$$

Then we incorporate the MIS, in which p_i^j represents the PDF of the i-th sample using j-th sampling strategy, and w_i^j is its corresponding MIS weight:

$$\partial_{\mathbf{m}} \int I dx \approx \partial_{\mathbf{m}} (\frac{1}{N} \sum_{j}^{M} \sum_{i}^{N} w_i^j(\mathbf{m}) \frac{g(\mathbf{m})}{p_i^j(\mathbf{m})}). \tag{4.10}$$

Finally, we approximate the gradient by following the *detached* sampling strategy:

$$\partial_{\mathbf{m}} \int I dx \approx \frac{1}{N} \sum_{j}^{M} \sum_{i}^{N} w_i^j(\mathbf{m}) \frac{\partial_{\mathbf{m}} g(\mathbf{m})}{p_i^j(\mathbf{m})}. \tag{4.11}$$

We bypass the computation of the gradients of the MIS weights $w_i^j(\mathbf{m})$ and the sampling PDF $p_i^j(\mathbf{m})$, and solely back-propagate gradients derived from $\partial_{\mathbf{m}} g(\mathbf{m})$ to the scene parameters $\mathbf{m}$ instead.

4.5 Optimization

We supervise the model only through an L2 loss between the rendering C and reference images C_R:

$$Loss = \|C - C_R\|_2^2 \tag{4.12}$$

For each interaction, we randomly select N_{batch} pixels from an image to serve as the model input. Our model then generates N_{SPP} rays for each pixel, which point from the camera's center towards the pixel area, using stratified sampling. If any of these rays intersect with the object's surface, we assess and record the scattering properties. We then use the BRDF sampling to determine new directions for each ray. Next, We repeat the intersection test and the succeeding process. For each bounce, an additional ray is introduced using emitter sampling. If a ray exits the scene, we query the emitted radiance from the environment in the ray's direction and compute its reflected radiance along its trajectory, and integrate it with MIS weights. This recursive process stops once it reaches the maximum bounce count, denoted as N_{bounce}. Subsequently, we take a mean of the N_{SPP} rays and get the predicted color. We then determine the loss by comparing the predicted color with the ground truth, and the gradient is back-propagated to both the BRDF and the environment.

5 Experiments

Dataset To comprehensively evaluate the performance of our model, we conduct experiments on two synthetic datasets and one real-world dataset:

- **N-Synthetic dataset** stands for the NeRFactor-Synthetic dataset [Zha+21b], which is rendered using Blender. The dataset comprises of four synthetic scenes: ficus, Lego, drums, and hotdog, all adapted from the NeRF dataset [Mil+20]. Each scene contains 100 training views, 8 validation views, and 200 test views. In particular, the dataset offers multiple images under varying lighting conditions, which can serve as a ground truth benchmark for assessing the relighting performance of the model. The ground truth images have 512x512 resolution and the environment maps have 32x64 resolution.
- **T-Synthetic dataset** stands for TensoIR-Synthetic dataset [Jin+23]. Similar to N-Synthetic dataset, it features four synthetic scenes: ficus, Lego, armadillo, and hotdog, each rendered at an elevated resolution of 800x800. Additionally, the environment maps are more detailed with a resolution of 1024x512.
- **DTU** [Jen+14] is a large-scale real-world dataset for multiple view stereo evaluation. It consists of diverse scenes with multi-view images and precise camera calibrations. We constrain our evaluation on a subset of DTU.

We assess our model on both the N-synthetic and T-synthetic datasets to enable a fair comparison to baseline methods. The different scale of resolution in the datasets allows us to evaluate the model performance more comprehensively. For the real-world dataset, unfortunately, the DTU lacks images of scenes captured under multiple lighting conditions, limiting us to a qualitative performance evaluation.

Metrics We quantitatively assess performance of our model using three methods, which are commonly employed to evaluate the similarity between two images.

- **PSNR** represents the peak signal-to-noise ratio. PSNR is a traditional metric used to measure absolute pixel differences.

- **SSIM** stands for the structural similarity index measure, which focuses on structural, luminance, and contrast differences between images, providing a more perceptual comparison.
- **LPIPS**: represents the learned perceptual image patch similarity, introduced by [Zha+18]. LPIPS captures the perceptual similarity between two images using pre-defined neural networks and aligns more consistently with human visual evaluations.

Implementation details We train our model on 100 training images for the synthetic datasets and use images from all 49 viewpoints for the DTU dataset. The ADAM optimizer [KB15] is employed with a learning rate set at 0.005 and a maximum of 70 epochs for every experiment. We use a batch size of 4096, i.e. randomly sampled 4096 pixels as input for the model. Unless otherwise specified, both training and testing utilize 512 SPP and a maximum bounce count of 2. All experiments are executed on a single RTX3070Ti GPU with 8 GB of memory. To address the ambiguity between lighting and object color, we scale each RGB channel of every albedo, following [Zha+21b; Jin+23]. Each scene takes approximately 40 minutes for training. For inference, images for 200 views of size 512x512 require about 25 minutes, while 200 views of size 800x800 take roughly 45 minutes.

5.1 Synthetic Dataset Results

In this section we conduct a comparison between our model, NeRFactor [Zha+21b], and the state-of-the-art model TensoIR [Jin+23]. As illustrated in Table 5.1, while our model may not surpass TensoIR in albedo estimation and novel view synthesis, it presents comparable or better relighting performance. This underscores the robustness of our model in the relighting tasks, which is attributed to its precise rendering pipeline and superior material decomposition.

N-Synthetic Dataset									
	Albedo			Novel View Synthesis			Relighting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRFactor	24.202	0.925	0.072	22.759	0.911	0.078	22.820	0.892	0.089
TensoIR	26.330	0.937	0.048	32.331	0.968	0.023	24.400	0.918	0.062
Ours	22.364	0.894	0.078	30.145	0.952	0.029	25.530	0.924	0.048
T-Synthetic Dataset									
	Albedo			Novel View Synthesis			Relighting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRFactor	25.125	0.940	0.109	24.679	0.922	0.120	23.383	0.908	0.131
TensoIR	28.512	0.952	0.043	35.046	0.975	0.019	28.533	0.944	0.046
Ours	24.578	0.898	0.057	30.582	0.938	0.028	25.888	0.911	0.046

Table 5.1: **Quantitative comparisons on N-Synthetic and T-Synthetic dataset results.** Our model outperforms baseline methods in relighting tasks on the N-Synthetic dataset. Although it exhibits lower proficiency in albedo reconstruction and novel view synthesis, its superior relighting results suggest a more accurate rendering pipeline and enhanced material property decomposition in our model. On the T-Synthetic dataset, which features larger image sizes, our model does not exceed the baseline in relighting, but maintains a good score in LPIPS, signifying its ability to produce relighting results that are perceptually realistic.

Scene decomposition and novel view synthesis As shown in Figure 5.1 and Figure 5.2, our model can disentangle the scene into albedo and lighting with high quality. The decomposed albedo embeds less lighting and shadows. Unlike the baseline methods that use handcrafted albedo smoothness to reduce embedded lighting, our model

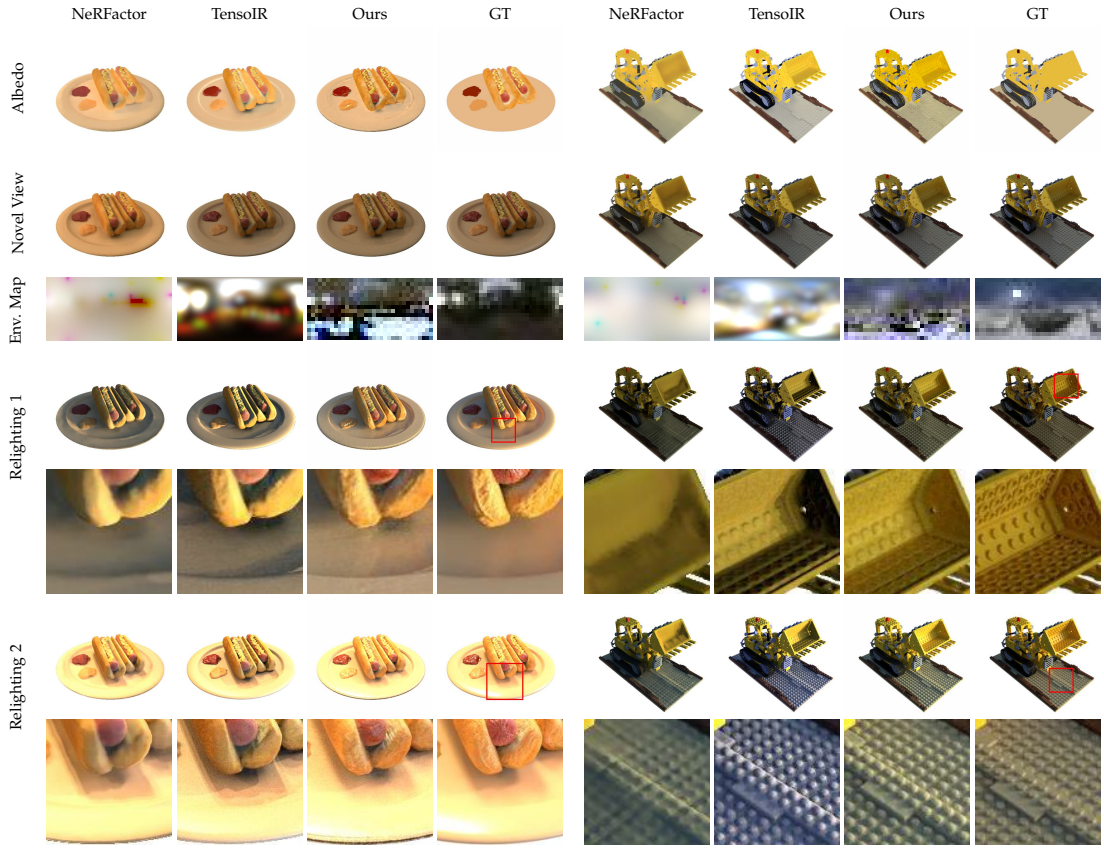

Figure 5.1: **Results comparison on N-Synthetic dataset.** Our model can factorize a scene into material properties and lighting. Our model produces high-quality relighting results characterized by realistic self-reflection, enhanced shadow fidelity under indirect illumination, natural soft shadows, and accurate color prediction.

achieves consistent albedo by simulating the real-world multiple light path bounces. Additionally, light sources are correctly recognized in the environment maps. Figure 5.4 and Figure 5.5 further demonstrate the high-quality of the decomposed BRDF parameters for additional scenes.

The primary reason for a lower metric score for albedo and novel view synthesis lies in the quality of the geometry. TensoIR queries precise surface locations from its implicit density field. In contrast, our model relies on meshing geometry extracted from TensoIR, which contains enormous artifacts, as depicted in Figure 4.2. The geometry quality significantly influences the performance of our model. However, with

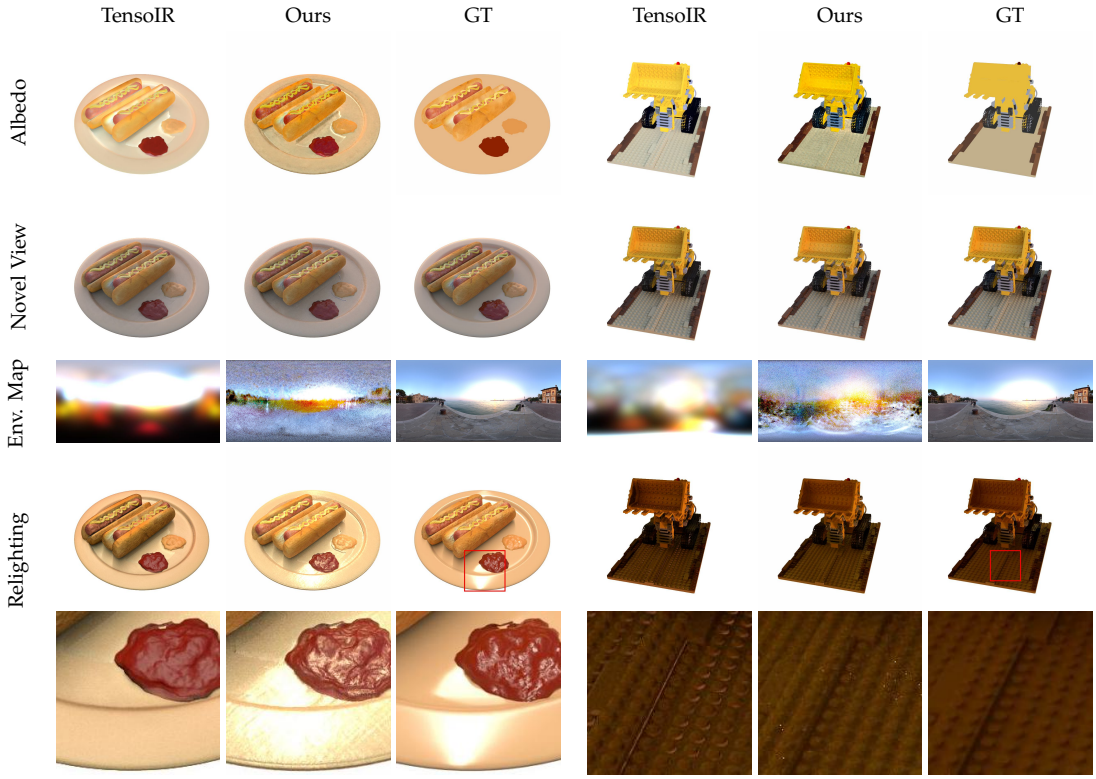

Figure 5.2: **Results comparison on T-Synthetic dataset.** Similar to the results on N-Synthetic shown in Figure 5.1, Our model produces high-quality relighting results characterized by improved specularity performances, and the elimination of the overcompensation issue of the pre-optimized indirect illumination estimation (See Figure 5.10 for more details).

suboptimal geometry, our model still delivers impressive relighting outcomes.

Relighting While our model may not excel in albedo estimation and the novel view synthesis task, it surpasses baseline models in the relighting task on the N-Synthetic Dataset. Though albedo serves as a metric for assessing material decomposition quality, it primarily reflects the quality of diffuse reflection. On the other hand, a high score in the relighting task truly signifies the model performance for material estimation. On the N-Synthetic Dataset, our model achieves the highest score in relighting, which demonstrates a more accurate material decomposition capability.

As illustrated in Figure 5.1 and Figure 5.2, our model offers significant advantages in relighting. Firstly, it adeptly reconstructs secondary shading effects. For instance, in

the hotdog scene, the self-reflection of the hotdog on the plate is discernible. Similarly, in the Lego scene, shadows in the digging bucket appear brighter and more realistic compared to TensoIR due to the accurate simulation of indirect illumination. Our model produces natural soft shadows by effectively sampling the low-resolution environment map. In addition, the decomposition of material properties is more precise, leading to a realistic reconstruction of base color and highlights. Our model also eliminates the overcompensation issue. For more details on this subject, please refer to Figure 5.10

On the T-synthetic dataset, our model faces two primary challenges in relighting. First, as image size grows, geometric artifacts become more prominent. Second, TensoIR tends to produce better results given a higher-resolution environment map for relighting. This is because TensoIR derives irradiance directly from the nearest point to the sample point. Thus, it can only generate natural soft shadows with sufficiently high-resolution environment maps, which is the case for T-synthetic. In contrast, our model can handle a low-resolution environment map by taking the weighted sum of irradiance around the sample point. In summary, the strengths of our model diminish as the shortcomings of geometry meshing intensify. Despite this, our model still achieves the highest LPIPS score.

Material editing Figure 5.3 demonstrates the material editing results. As our model decomposes the BRDF parameters explicitly, modification of the material properties is feasible, such as changing the diffuse base color or adjusting the specularity. Notably,

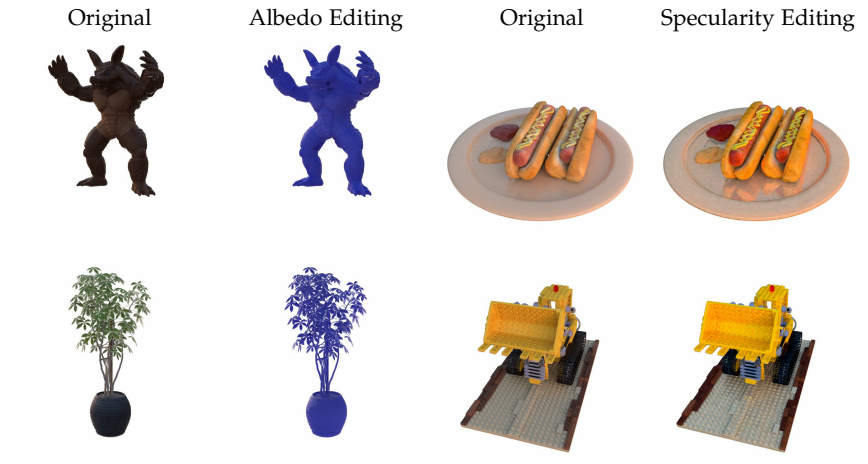

Figure 5.3: **Results of Material Editing.** Our model enables editing of albedo and specularity of the scene. The online multi-bounce computation of our model facilitates the prediction of realistic self-reflection.

the edited scene adeptly generates physically based secondary shading effects like self-reflection.

N-Synthetic Dataset

Scene	Method	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	NeRFactor	29.029	0.948	0.051	21.889	0.937	0.075	25.589	0.915	0.084
	TensoIR	28.786	0.949	0.047	37.736	0.980	0.015	26.122	0.912	0.075
	Ours	28.261	0.927	0.056	34.053	0.955	0.034	28.225	0.918	0.056
Lego	NeRFactor	24.898	0.926	0.075	25.193	0.886	0.087	24.361	0.856	0.097
	TensoIR	24.766	0.922	0.059	34.201	0.961	0.019	25.335	0.906	0.055
	Ours	24.237	0.875	0.065	30.974	0.944	0.019	27.468	0.917	0.031
Ficus	NeRFactor	21.888	0.925	0.076	20.640	0.909	0.081	20.706	0.908	0.083
	TensoIR	26.853	0.960	0.033	29.222	0.971	0.031	24.120	0.942	0.058
	Ours	20.552	0.917	0.064	28.192	0.961	0.034	23.486	0.939	0.054
Drums	NeRFactor	20.991	0.900	0.088	23.312	0.910	0.071	20.625	0.887	0.092
	TensoIR	24.914	0.918	0.053	28.164	0.960	0.026	22.021	0.914	0.061
	Ours	16.405	0.855	0.129	27.360	0.947	0.029	22.906	0.914	0.050

T-Synthetic Dataset

Scene	Method	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	NeRFactor	24.654	0.950	0.142	24.498	0.940	0.141	22.713	0.914	0.159
	TensoIR	26.897	0.957	0.046	36.722	0.975	0.018	27.790	0.932	0.063
	Ours	26.494	0.920	0.066	34.092	0.949	0.035	28.085	0.918	0.067
Lego	NeRFactor	25.444	0.937	0.112	26.076	0.881	0.151	23.246	0.865	0.156
	TensoIR	25.260	0.924	0.057	34.707	0.968	0.016	27.757	0.923	0.049
	Ours	21.078	0.807	0.074	28.310	0.886	0.030	25.162	0.863	0.044
Ficus	NeRFactor	22.402	0.928	0.085	21.664	0.919	0.095	20.684	0.907	0.107
	TensoIR	27.267	0.965	0.032	29.752	0.973	0.029	24.265	0.946	0.053
	Ours	21.553	0.930	0.051	27.504	0.955	0.034	21.048	0.921	0.051
Armadillo	NeRFactor	28.001	0.946	0.096	26.479	0.947	0.095	26.887	0.944	0.102
	TensoIR	34.624	0.960	0.039	39.003	0.985	0.013	34.322	0.975	0.019
	Ours	29.187	0.933	0.037	32.424	0.964	0.014	29.256	0.942	0.021

Table 5.2: **Per-scene quantitative results.**

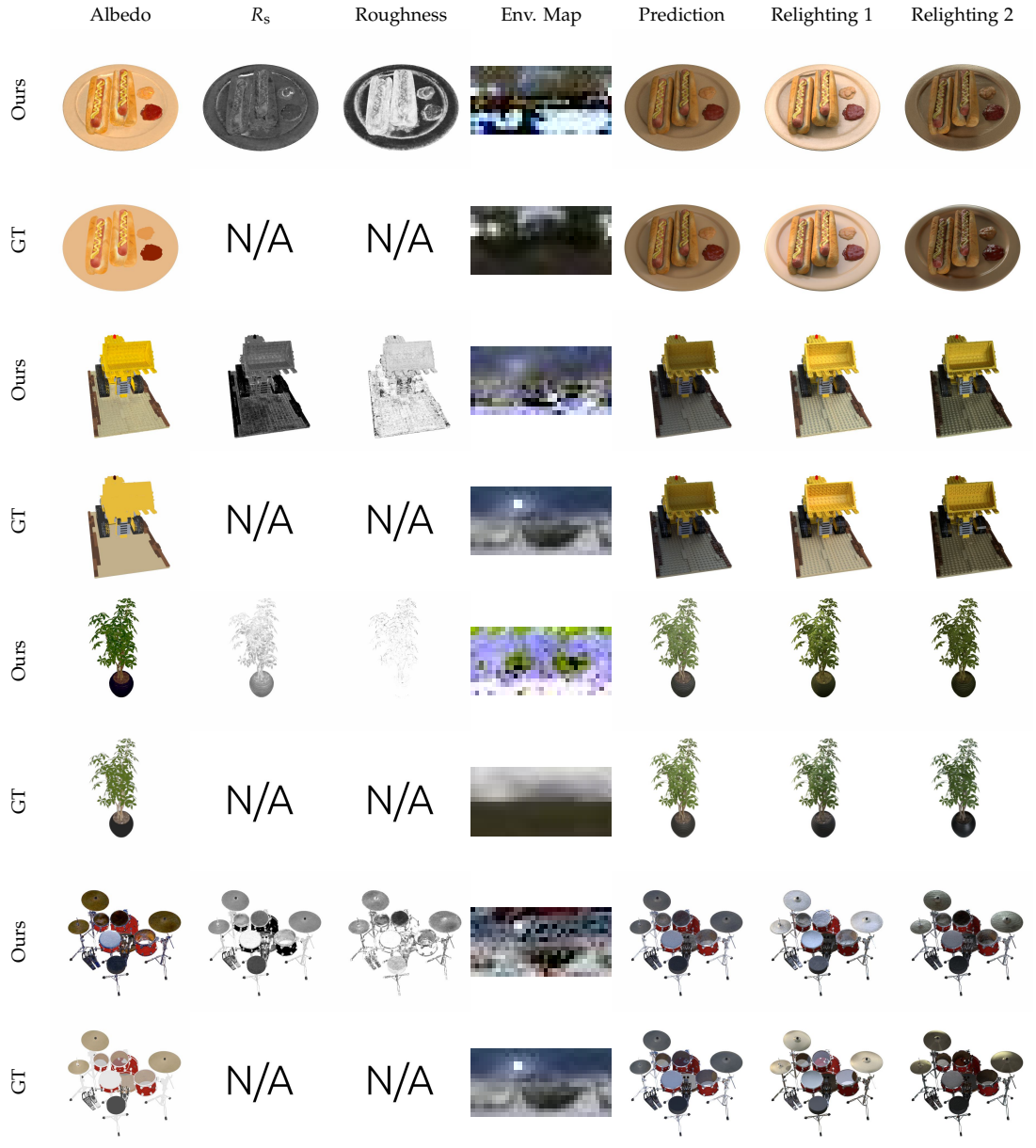

Figure 5.4: **Results of material and lighting decomposition, novel view synthesis, and relighting on N-Synthetic dataset.** Note that the dataset lacks ground truth visualizations for specular reflectance R_s and roughness.

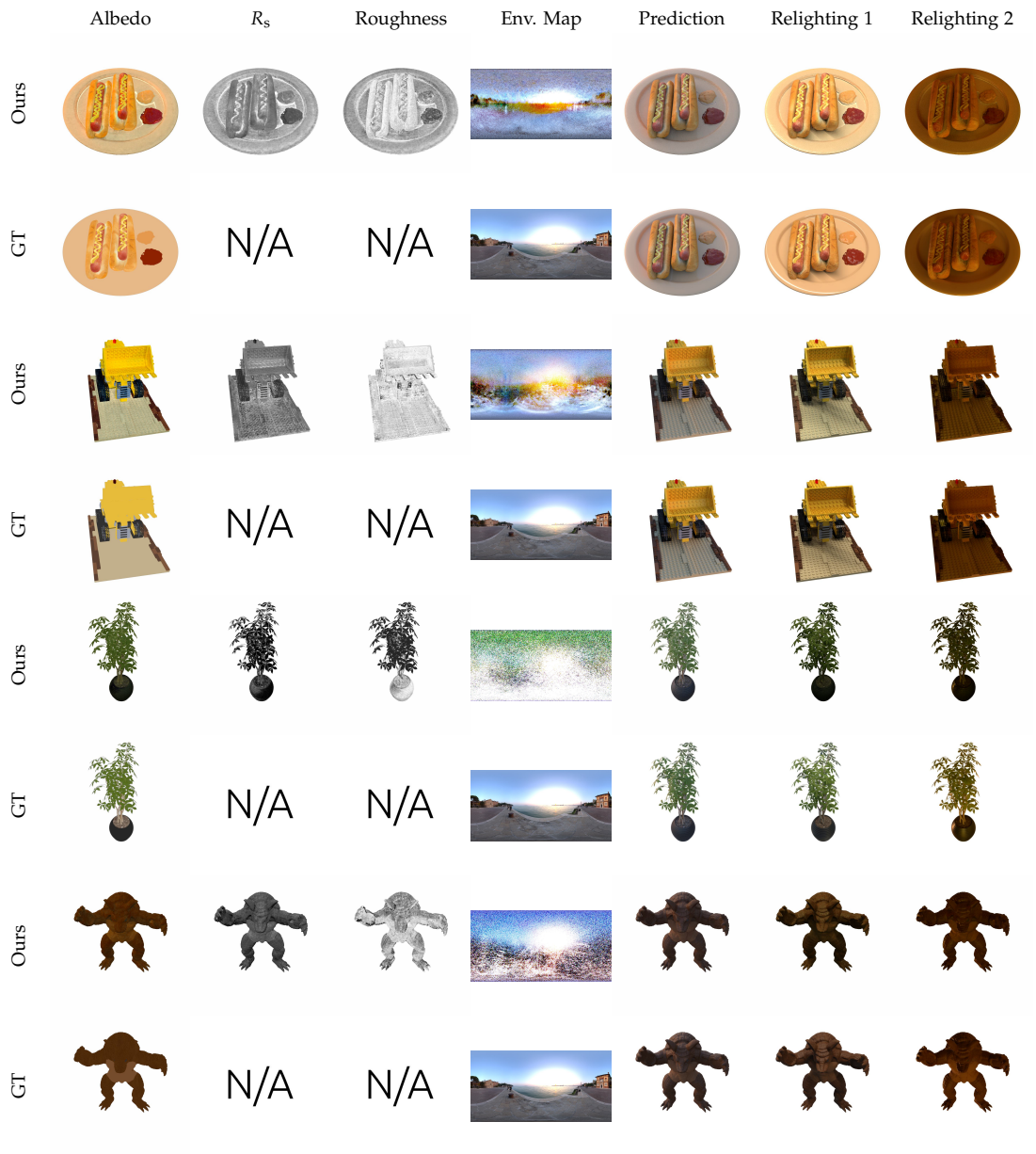

Figure 5.5: **Results of the material and lighting decomposition, novel view synthesis, and relighting on T-Synthetic dataset.** Note that the dataset lacks ground truth visualizations for specular reflectance R_s and roughness.

5.2 Real World Dataset Results

We also evaluate the model performance on real-world scenes. As Figure 5.6 shows, the appearance is factorized into material properties and lighting. Our model predicts reasonable albedo and roughness of the objects. Even though the GT albedo is not accessible in this datasets, our model is robust to decompose qualitatively accurate albedo, without using GT albedo to conduct the absolute brightness correction. For rough surfaces, our model correctly predicts a low R_s value, which corresponds to the absence of specular reflection. On the other hand, for specular surfaces, our model delievers a low roughness value with reasonable corresponding R_s value. The lower parts of environment maps are not optimized because objects are placed on a large table, and thus few rays can arrive at the lower hemisphere of the environment.

Figure 5.7 visualize the remarkable relighting results. In general, our relighting results feature reasonable base colors, adaptable specular effects based on lighting conditions, and realistic shadows. Our model effectively reconstructs the specular reflection on the object surface, with the specular highlights being varied under different lighting conditions. A further standout feature is the physically correct shadow estimation. Our model successfully reconstructs realistic soft and hard shadows.

A significant challenge is acquiring precise and clean geometry. When provided with images predominantly from one angle, such as the front side of an object, the reverse side of the geometry might exhibit considerable artifacts. These artifacts can significantly influence performance because we assume that the scene is illuminated solely by an ambient light source around the object. If the light path is obstructed by these artifacts, it can lead to unrealistic shadows in our results, even if the artifacts themselves are not visible in the rendered images.

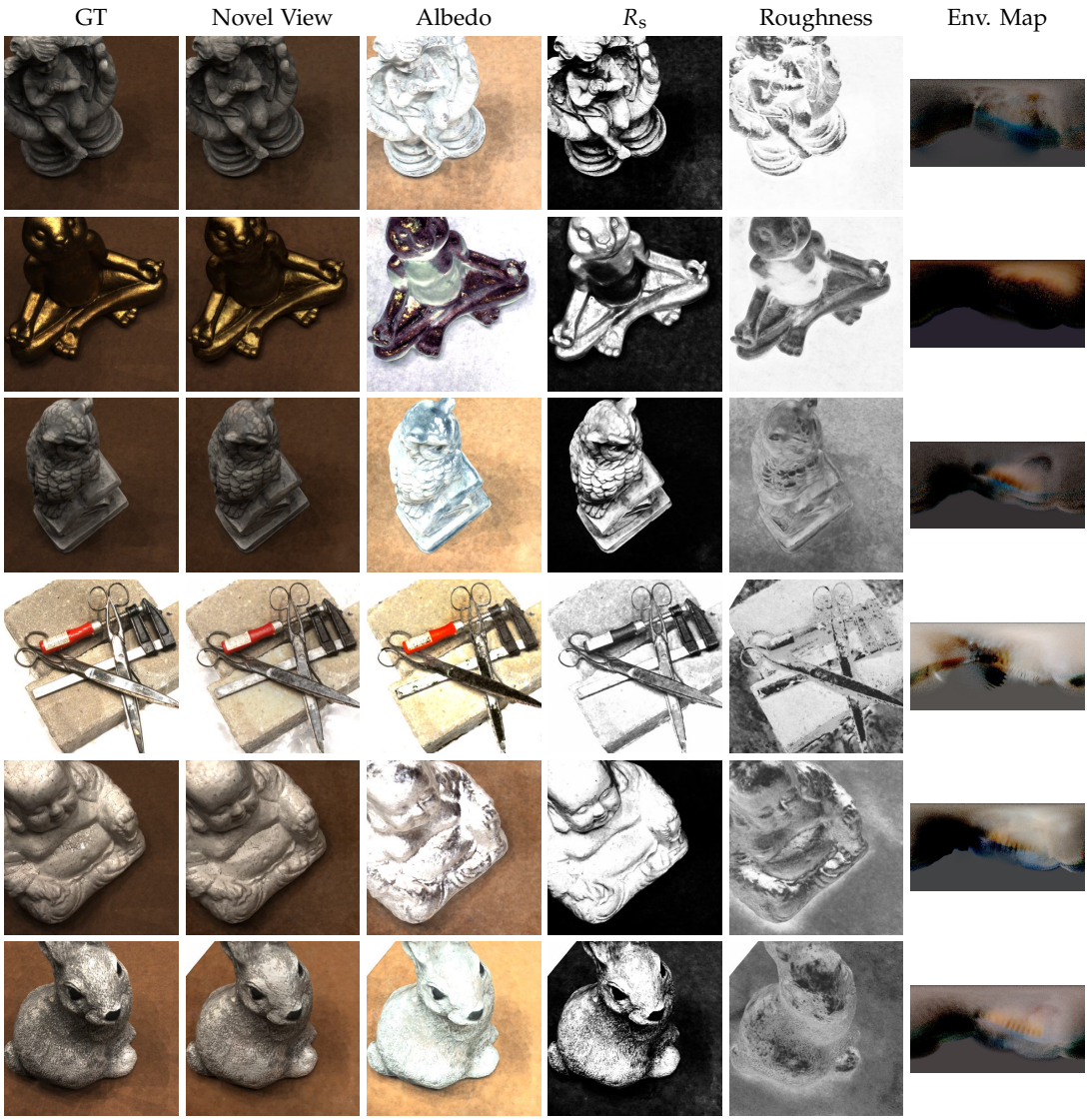

Figure 5.6: **Results of material and lighting decomposition, and novel view synthesis on the DTU dataset.** Our model can disentangle reasonable material properties and lighting from real-world scenes.

Figure 5.7: **Results of relighting on DTU dataset.**

5.3 Analysis of Multi-Bounce Global Illumination Estimation

In this section, we discuss the benefits and drawbacks of incorporating multi-bounce global illumination estimation in neural rendering. To eliminate the influence of noisy geometry, all experiments consider GT geometry. The impact of geometry is discussed in Subsection 5.4.1.

5.3.1 Beneficial impact on rendering.

In computer graphics, an increased number of bounces often correlates with more realistic rendering. As shown in Figure 5.8, our model adeptly renders the images for each individual light path bounce with the final output as their cumulative result. The contribution of the first bounce is referred to as direct illumination. Subsequent bounces, starting from the second, are classified as indirect illumination.

It is evident that direct illumination dominantly determines the quality of the output images. The second bounce contributes to the most part of secondary effects like self-reflection and enhanced brightness in the shadow area. From the third bounce onward, the contributions diminish. This pattern holds for distinct scenes, though the speed of diminishing varies with respect to the surface material. As material specularity or shape complexity increases, the influence of indirect illumination becomes more pronounced. In such cases, a thorough analysis of indirect illumination becomes important. Notably, even in predominantly diffuse scenes, the contribution of indirect illumination is not neglectable. The estimation of multi-bounce reflection contributes to the decomposition of material properties during training and the estimation of secondary shading effects during inference.

5.3.2 Beneficial impact on material decomposition.

The ability to factorize the contribution from each light bounce enables a cleaner albedo reconstruction, in which less lighting and shadow get embedded. Figure 5.9 and Table 5.3 demonstrate qualitatively and quantitatively how the reconstructed albedo quality changes with the number of maximal bounces. It is evident that the performance increases significantly as the number of maximum bounces rises.

The single-bounce result represents the reconstructed albedo for methods like NeRFactor, which neglect indirect illumination. Through analyzing sorely the direct illumination, lighting and shadows get baked into the albedo. The reason for this is that the self-occlusion shadows often receive indirect illumination which is generated by the reflection of its surroundings. Consequently, these shadows appear brighter than if only direct illumination are considered. Those brighter shadows get baked into albedo

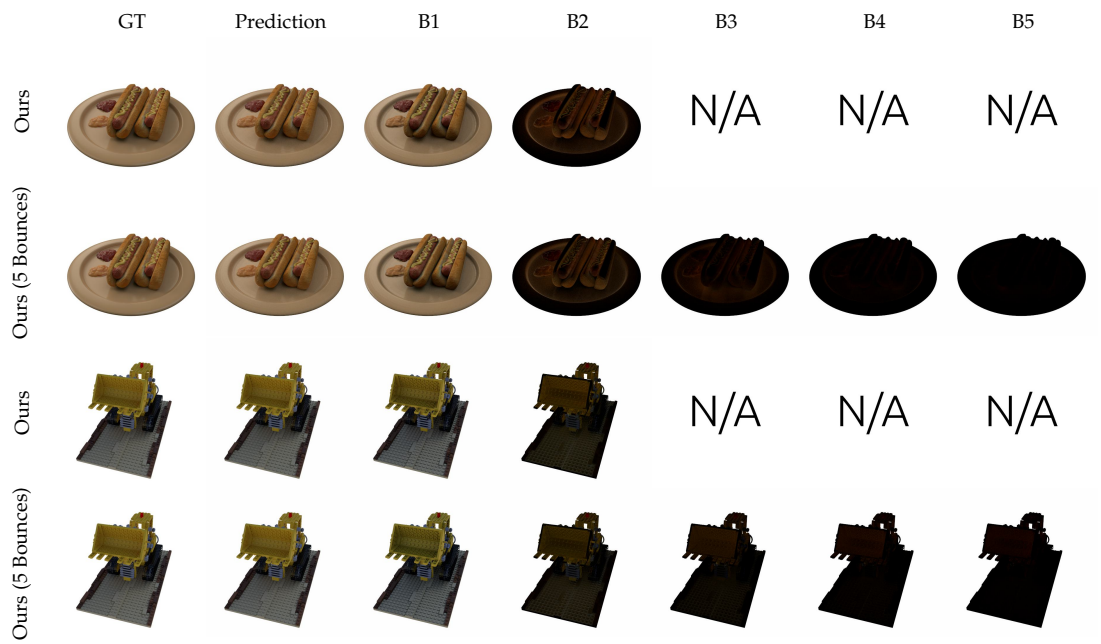

Figure 5.8: **Visualization for the contribution from each individual light path bounce.** B1 represents the first bounce, commonly referred to as direct illumination. Starting from the second bounce B2, it is termed as indirect illumination. In our case, contributions from the third bounce and beyond become barely perceptible, because the scenes predominantly consist of diffuse materials.The default maximal number of bounces for our model is set to 2, therefore its visualizations for B3, B4, and B5 are absent.

if only direct illumination is in consideration.

To address this issue, baseline methods rely on an albedo smoothness loss to give prior knowledge, which implies that GT albedo should contain less variance within a small region. However, this does not always hold for distinct real-world scenarios. Such priors can result in over-smoothed surface details. Our model, in contrast, addresses this issue through a physically-based multi-bounce global illumination estimation without any smoothness assumption. Multi-bounce global illumination enables our model to reconstruct accurate material properties without relying on additional prior assumptions. In addition, we also investigate the impact of smoothness on our model in Subsection 5.4.3, in which we come to the conclusion that incorporating a smoothness assumption does not enhance the performance in general.

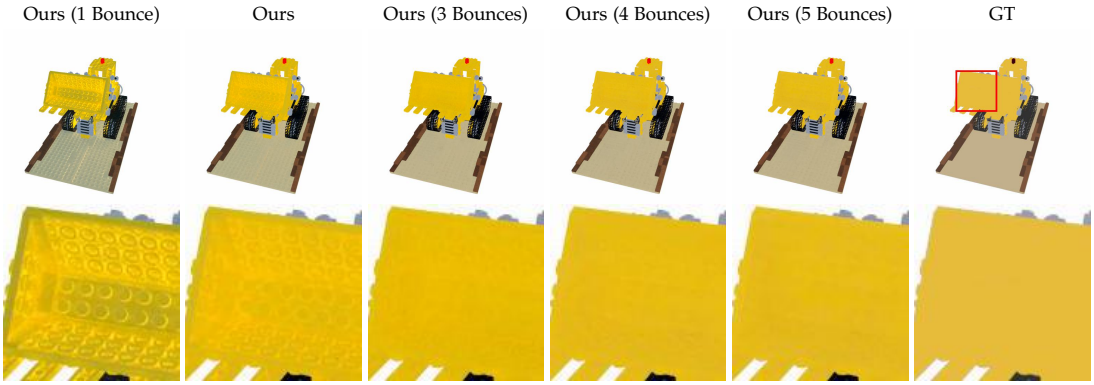

Figure 5.9: **Estimation of the impact of multi-bounce on albedo reconstruction.** With an increase in the number of light path bounces, lighting and shadows that get baked into the albedo diminish.

Scene	Max. Bounce	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	1	28.518	0.929	0.062	37.986	0.970	0.017	30.276	0.934	0.044
	2	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
	3	30.659	0.946	0.034	39.086	0.975	0.013	30.887	0.940	0.038
	4	30.907	0.947	0.034	39.046	0.975	0.014	30.862	0.940	0.037
	5	31.023	0.948	0.033	39.061	0.975	0.014	30.857	0.940	0.038
Lego	1	25.369	0.900	0.055	36.526	0.974	0.008	29.156	0.961	0.023
	2	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010
	3	25.191	0.939	0.020	37.451	0.980	0.005	33.370	0.974	0.008
	4	27.051	0.956	0.017	37.589	0.980	0.005	33.294	0.974	0.008
	5	26.935	0.956	0.017	37.616	0.980	0.005	33.307	0.974	0.008

Table 5.3: **Quantitative comparison of the effect of maximum light bounces.** In general, as the number of light bounces increases, so does the performance in albedo, novel view synthesis, and relighting. Notably, the jump from single bounce to two bounces achieves most significant improvement.

5.3.3 Comparison against the pre-optimized indirect illumination estimation

In comparison against the pre-optimized and fixed indirect illumination estimation, our multi-bounce path tracer performs an online indirect illumination estimation. For any given lighting condition, our model can estimate physically based secondary shading effects. Some methods (e.g. TensoIR) estimate indirect illumination by querying a radiance field along the second bounce direction. The estimated indirect illumination is learned from the training scene, and can not be modified during scene editing. As

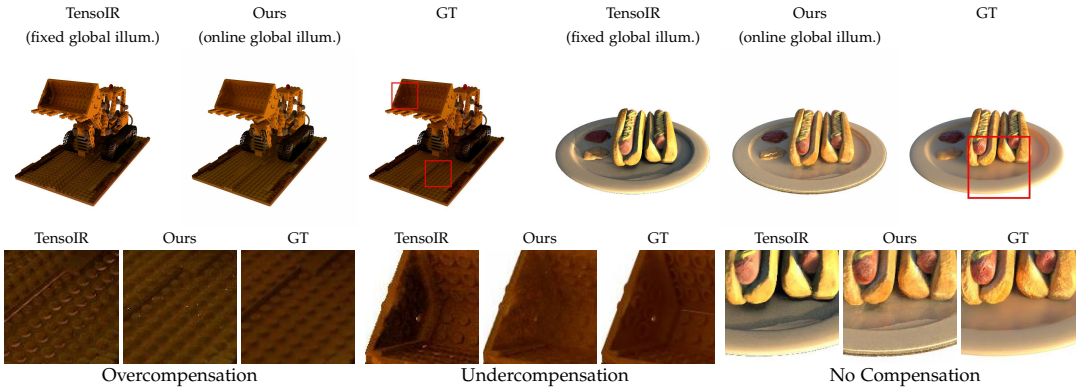

Figure 5.10: **Comparison of relighting using fixed vs. online global illumination estimation.** Our model, leveraging online-calculated global illumination estimation, demonstrates enhanced performance compared to models using pre-optimized fixed global illumination estimation. The fixed approach presents three evident issues: overcompensation, where the indirect illumination is excessively strong; undercompensation, with notably weak indirect illumination; and no compensation, indicating a total absence of indirect illumination.

a result, those methods struggle at estimation of indirect illumination under novel lighting conditions. In Figure 5.10, we visualize three kinds of issues of using a pre-optimized global illumination estimation:

- **Overcompensation issue**: This refers to an excessively strong indirect illumination effect. For instance, the relighting result from TensoIR has unnatural highlights on the side of the Lego studs. This issue arises because TensoIR has been optimized based on a training scene with significantly brighter lighting. Consequently, TensoIR remembers the strong indirect illumination. When given a darker novel lighting scenario, this pre-optimized indirect illumination appears overly bright, leading to noticeable bright edges. In contrast, our model can correctly simulate the indirect illumination given any novel lighting , thus our results align more closely with the GT.

- **Undercompensation issue**: In contrast to the overcompensation issue, undercompensation signifies a distinctly weak indirect illumination effect. For instance, within the Lego scene, it is evident that the shadow in the digging bracket of TensoIR results is noticeably darker. Although TensoIR learned part of the indirect illumination, it is not sufficient to represent the actual indirect illumination. Our

model inherently solved this issue through the multi-bounce global illumination estimation.

- **No compensation**: As an extreme case of the undercompensation issue, some secondary effects such as self-reflection are completely absent in TensoIR. Conversely, our model adeptly simulates inter-object light reflection, yielding renderings with realistic self-reflection.

5.3.4 Drawbacks of multi-bounce light path estimation

While multi-bounce global illumination offers numerous benefits, it also presents certain drawbacks. These drawbacks can be broadly categorized into two main areas: quality-related issues and cost-related issues.

In some cases, we observe that more bounces may decrease the quality of rendering. As shown in Table 5.3, the PSNR for relighting achieves the highest score at two bounces. It is evident that transitioning from one to two bounces significantly enhances the performance, therefore leading to a better PSNR score. However, as the number of bounces further increases, relighting results become worse in both hotdog and Lego scenes. Since the SSIM and LPIPS scores remain a high standard, we suggest that the reason should be the accumulated numerical errors. Figure 5.11 illustrates the 'fireflies', which are noises generated by MC. This often happens when there is a high variance in the radiance of the environment map. With more bounces, more numerical errors are generated, all accumulated to the final results, thus leading to decreases in the absolute pixel radiance similarity.

Another issue is cost related. In Figure 5.12, we demonstrate a relative comparison of training time and GPU memory usage. For reference, the model with two bounces is set as baseline, represented by 100%. We observe that the training time increases markedly with more bounces. Specifically, for 5 bounces, the training time was nearly three times compared to that of 2 bounces (2 hours versus 40 minutes). The choice of bounces represents a trade-off between time and model performance. Given that the majority of scenes in our datasets are characterized by diffuse materials, we select 2 bounces for our final model to optimize cost-effectiveness. The GPU memory usage doesn't substantially increase with added bounces. That enables training with larger SPP or using a larger batch size.

Figure 5.11: **'Fireflies' in path tracing.** Rendering noise, such as "fireflies", becomes more pronounced with more bounces, leading to a decline in rendering quality.

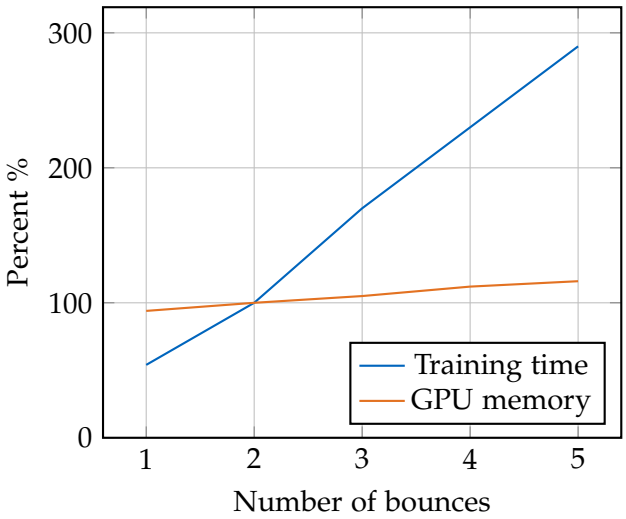

Figure 5.12: **Runtime and GPU memory usage.** The model with two bounces is set as the baseline, represented by 100%.

5.4 Ablation Study

5.4.1 Geometry

In this section, we explore the influence of geometry on the performance of our model. Our model employs a pre-optimized mesh representation that contains significant artifacts. To eliminate the negative effects of this noisy geometry, we extract the GT scene mesh directly from its Blender file. This experiment not only showcases the optimal performance of the model under ideal conditions but also underscores the impact of utilizing estimated geometry.

As shown in Figure 5.13, our model delivers superior results when provided with GT geometry. When training with estimated geometry, our model reconstruct albedo and roughness with noticeable noise, and lighting and shadows being embedded into these parameters. The use of GT geometry substantially reduces the noise without any smoothness assumptions, which results in more consistent albedo and roughness values. Notably, the specularity is sensitive to the geometry in some scene, such as in the hotdog scene in the T-Synthetic dataset. Quantitative results in Table 5.4 confirm that utilizing GT geometry significantly enhances model performance across all tasks and scenes. The full results for the N-Synthetic and T-Synthetic datasets can be viewed in Figure 5.14 and Figure 5.15.

It is notable that an accurately decomposed material and lighting can guide the reconstruction of geometry, conversely. In our work, we do not back-propagate the gradient to the geometry due to time limitation. We propose that an accurate rendering pipeline, which incorporates physically-based global illumination, has the potential to aid in finer geometry reconstruction.

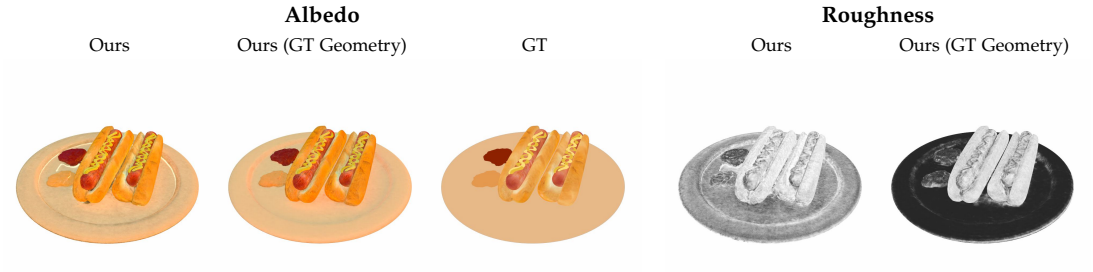

Figure 5.13: **Comparison of material decomposition using estimated vs. ground truth geometry.** Rendering using ground truth geometry yields significantly better performance in decomposing clean albedo and roughness.

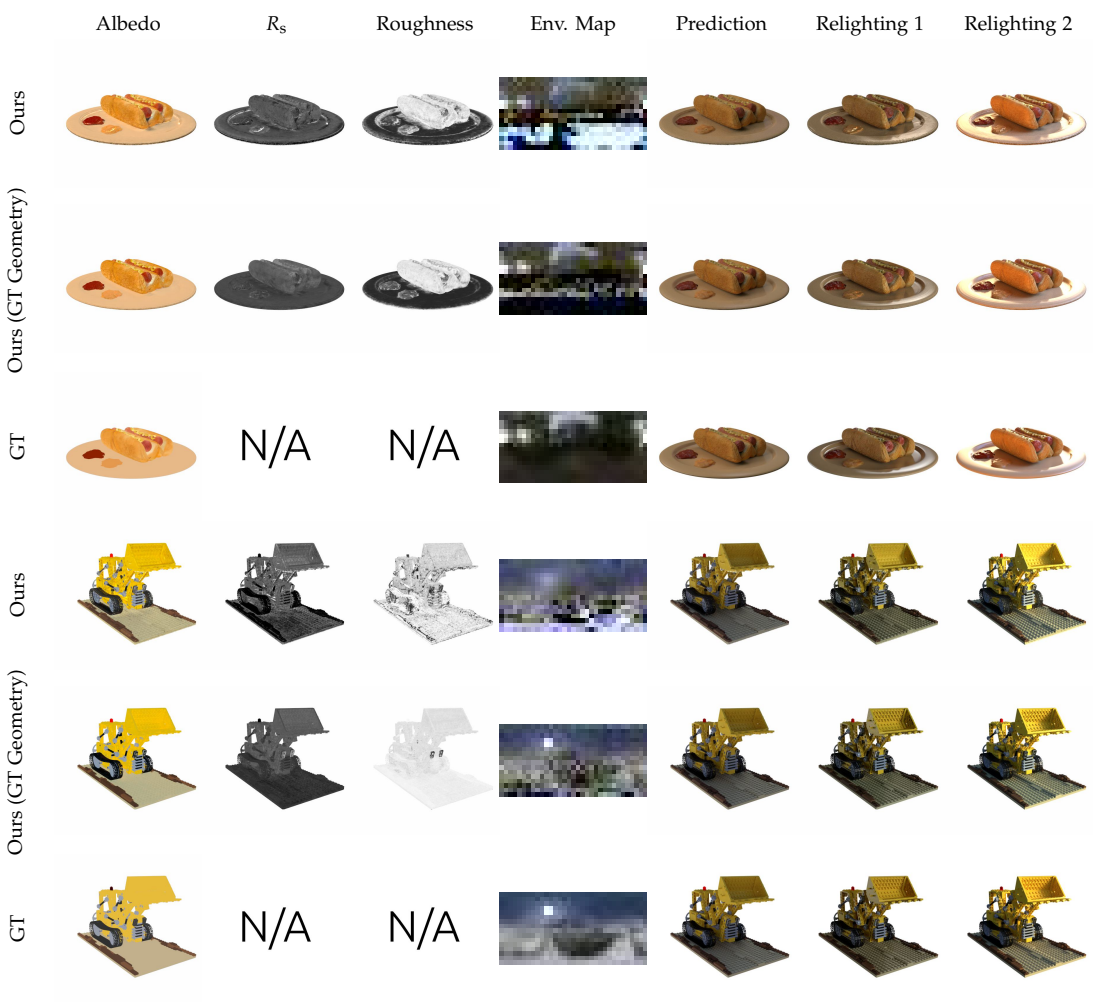

Figure 5.14: **Comparison of using estimated vs. ground truth geometry on N-Synthetic dataset.** The model with ground truth geometry yields significantly better performance across all tasks. Note that the dataset lacks ground truth visualizations for specular reflectance R_s and roughness.

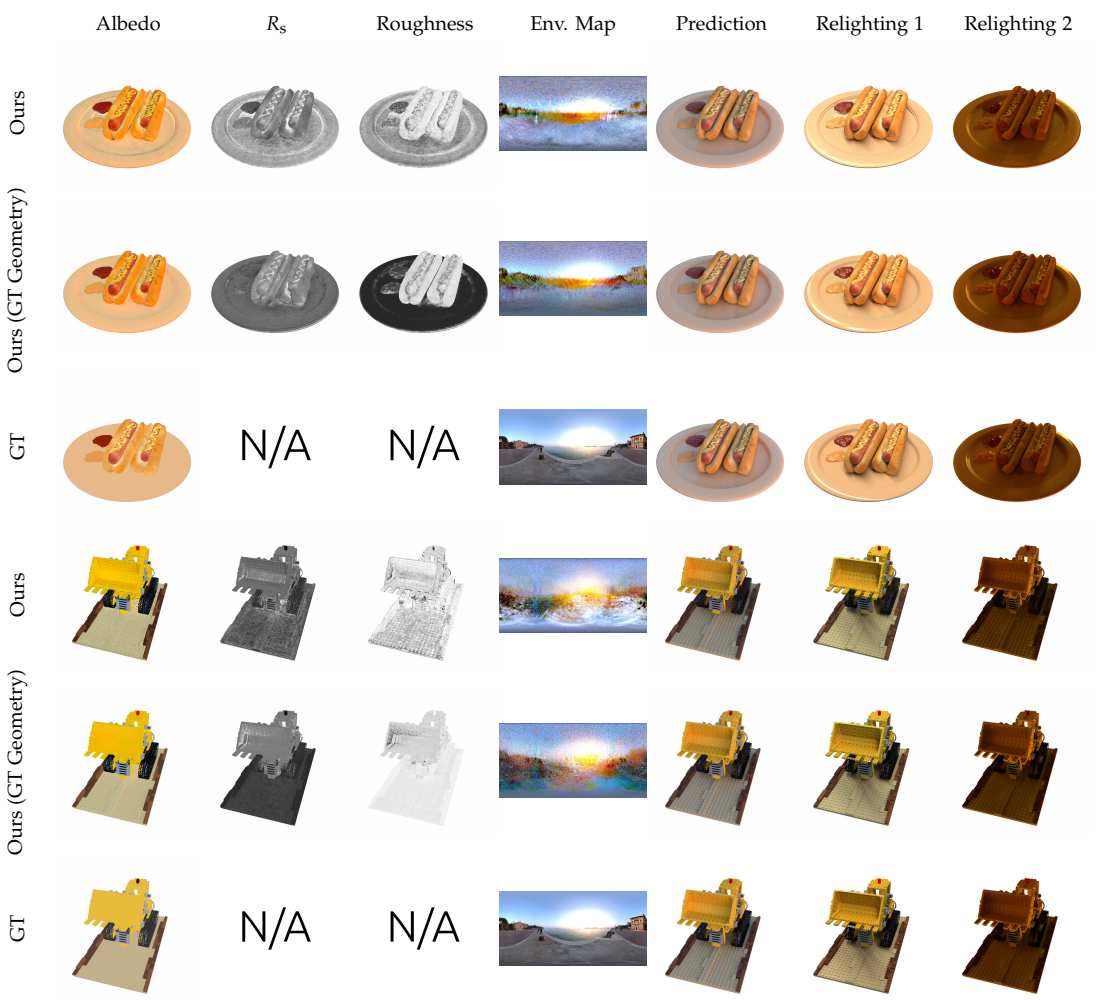

Figure 5.15: **Comparison of using estimated vs. ground truth geometry on T-Synthetic dataset.** The model with ground truth geometry yields significantly better performance across all tasks. Note that the dataset lacks ground truth visualizations for specular reflectance R_s and roughness.

N-Synthetic Dataset										
Scene	Geometry	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	Pretrained	28.766	0.921	0.060	34.114	0.954	0.036	28.014	0.914	0.059
	GT	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
Lego	Pretrained	24.072	0.875	0.065	30.844	0.944	0.020	27.516	0.917	0.033
	GT	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010
T-Synthetic Dataset										
Scene	Geometry	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	Pretrained	26.494	0.920	0.066	34.092	0.949	0.035	28.085	0.918	0.067
	GT	27.848	0.946	0.040	37.385	0.954	0.031	32.404	0.935	0.052
Lego	Pretrained	22.112	0.851	0.072	29.801	0.928	0.032	26.991	0.908	0.047
	GT	24.944	0.921	0.028	36.575	0.971	0.013	32.567	0.958	0.025

Table 5.4: **Quantitative comparison on geometries.** The model performance is strongly correlated with the quality of the geometry.

5.4.2 BRDF Model

In this section, we examine the effects of employing different BRDF models. Specifically, we compare the performance of the model when using solely the Lambertian acBRDF, the combination of Lambertian BRDF with the Phong reflectance model, and the Lambertian BRDF paired with the GGX microfacet reflectance model.

As illustrated in Table 5.5, the conclusion is straightforward: the model employing the GGX microfacet reflectance model surpasses its contestants. This model achieves top scores in both the novel view synthesis and relighting tasks, underscoring its superior material decomposition quality. A visual comparison in Figure 5.16 further demonstrates this point. The GGX model adeptly reconstructs an albedo without embedded lighting and accurately replicates a realistic specular reflection.

Two additional findings merit mentioning:

First, the model without specularity estimation achieves a remarkably high score in albedo estimation. This outcome is understandable as the model exclusively concentrates on diffuse reflection estimation. The albedo, in this context, precisely reflects the capacity of the model for estimating diffuse reflection. Another contributing factor could be related to sampling. In this case, sampling the BRDF is equivalent to sampling the diffuse lobe. Consequently, we employ cos-weighted sampling (together with emitter sampling). As we conclude in Subsection 5.4.5, cos-weighted sampling plays a pivotal role in enhancing the performance of albedo reconstruction.

Secondly, we observe a notable drop in the PSNR for albedo reconstruction in the Lego scene. Although the GGX and Phong models exhibit comparable relighting results, indicating similar material reconstruction capabilities, the albedo PSNR for the GGX model is significantly lower than the Phong model. This could be due to some diffuse reflection being misclassified as specular reflection in the GGX model. Notably, on very rough surfaces, the GGX microfacet performs similarly to diffuse reflection. These findings underscore that albedo, while informative, is not the sole metric for assessing material reconstruction quality. A more definitive evaluation would be the relighting results.

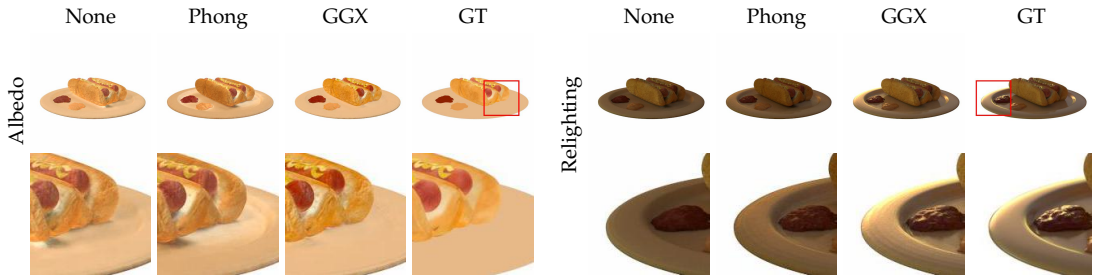

Figure 5.16: **Comparison on BRDF models.** Using the GGX microfacet model for specularity enhances the quality of both albedo and specular effects.

Scene	Spec. Model	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	None	30.667	0.947	0.041	35.786	0.973	0.017	28.853	0.934	0.047
	Phong	26.643	0.931	0.050	37.794	0.970	0.017	27.182	0.922	0.052
	GGX	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
Lego	None	32.356	0.965	0.023	36.581	0.977	0.007	32.482	0.968	0.013
	Phong	30.045	0.959	0.026	37.231	0.978	0.006	33.261	0.973	0.010
	GGX	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010

Table 5.5: **Quantitative comparison on BRDF models.** Using the GGX microfacet model for specularity achieves the best results in novel view synthesis and Relighting.

5.4.3 Smoothness

Baseline methods rely heavily on smoothness assumption. In this section, we investigate the impact of smoothness on the performance of our model. We follow the smoothness implementation in the baseline methods. For the high smoothness level, we set the smoothness weights $(0.5, 0.1, 0.1)$ for $(R_d, R_s, roughness)$. For the middle level and low level, the weights are $(0.05, 0.01, 0.01)$ and $(0.005, 0.001, 0.001)$, respectively.

The conclusion is that using smoothness does not improve the performance of our model. Table 5.6 illustrates a good example of the limitation of using smoothness. In the Lego scene, while smoothness increases the albedo PSNR, there is a marked drop in its LPIPS as well as the relighting metric score. This means smoothness only contributes to the smoothness of the absolute value difference, without taking human perception into account. Figure 5.17 further illustrates this effect: instead of refining the albedo, smoothness conversely introduces more artifacts.

Similar results can be observed in the case using estimated geometry, as demonstrated in Table 5.7: Once again, the albedo smoothness selectively enhances the albedo metric score while reducing the relighting performance. We conclude that the smoothness assumption does not enhance performanceof our model, given that our model, in its current form, already yields high-quality material reconstruction.

Scene	Smooth. Level	**Albedo**			**Novel View Synthesis**			**Relighting**		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	High	33.604	0.969	0.022	37.528	0.971	0.020	31.163	0.942	0.040
	Middle	33.033	0.964	0.022	38.796	0.975	0.015	31.871	0.943	0.036
	Low	32.487	0.961	0.024	38.837	0.974	0.015	31.604	0.943	0.036
	None	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
Lego	High	28.774	0.942	0.041	34.971	0.970	0.012	29.464	0.958	0.025
	Middle	24.123	0.920	0.030	36.405	0.976	0.008	31.203	0.964	0.017
	Low	23.995	0.920	0.026	36.684	0.978	0.006	32.712	0.971	0.012
	None	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010

Table 5.6: **Quantitative comparison on smoothness given ground truth geometry.** While integrating smoothness improves scores in albedo decomposition, it adversely affects the overall material estimation, resulting in a worse relighting performance.

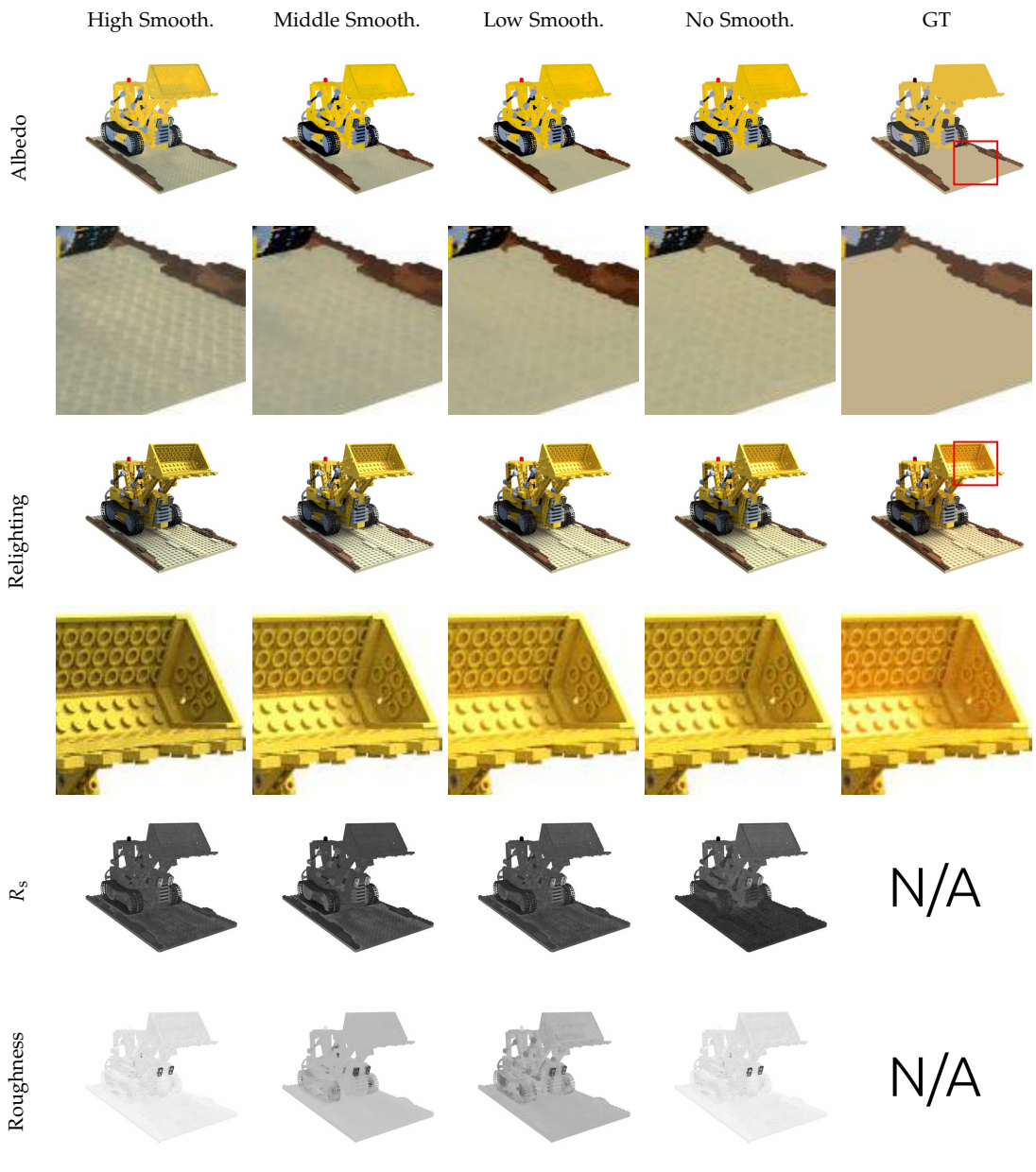

Figure 5.17: **Comparison of different smoothness.** The inclusion of smoothness damages the material decomposition in the hotdog scene (lighting and shadows get baked into albedo). The additional prior about smoothness assumption proved to be unnecessary in our model. Note that the dataset lacks ground truth visualizations for specular reflectance R_s and roughness.

	Albedo			Novel View Synthesis			Relighting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TensoIR (as Reference)	26.330	0.937	0.048	32.331	0.968	0.023	24.400	0.918	0.062
Ours (Smooth.)	23.222	0.908	0.072	30.000	0.951	0.030	25.089	0.918	0.054
Ours	22.364	0.894	0.078	30.145	0.952	0.029	25.530	0.924	0.048

Table 5.7: **Quantitative comparison on smoothness given estimated geometry.** Similar to the previous findings in Table 5.6, the introduction of smoothness hinders the relighting performance, indicating a negative impact on the overall material estimation.

5.4.4 Samples per Ray

In this section, we conduct an investigation of the impact of Samples Per Ray (SPP) during training. For a fair comparison, all images are rendered with same SPP in the test phase.

The conclusion is clear: higher SPP leads to better results. As demonstrated in Table 5.8, there is a clear trend of improved model performance across all tasks with increasing SPP. This improvement can be attributed to the fact that rendering with a lower SPP introduces noise, generating noisy gradients for training. Consequently, gradient descent methods converge to a suboptimum point. Figure 5.18 further emphasize the conclusion.

Scene	SPP	Albedo			Novel View Synthesis			Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	32	28.091	0.928	0.049	37.870	0.972	0.017	28.252	0.931	0.045
	128	29.673	0.941	0.036	38.804	0.975	0.014	30.917	0.939	0.039
	512	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
Lego	32	24.726	0.906	0.052	36.201	0.975	0.008	29.107	0.957	0.020
	128	24.985	0.924	0.028	37.160	0.979	0.006	32.423	0.970	0.012
	512	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010

Table 5.8: **Quantitative results of different SPP.** As the number of training SPP increases, our model demonstrates enhanced performance in albedo, novel view synthesis, and relighting.

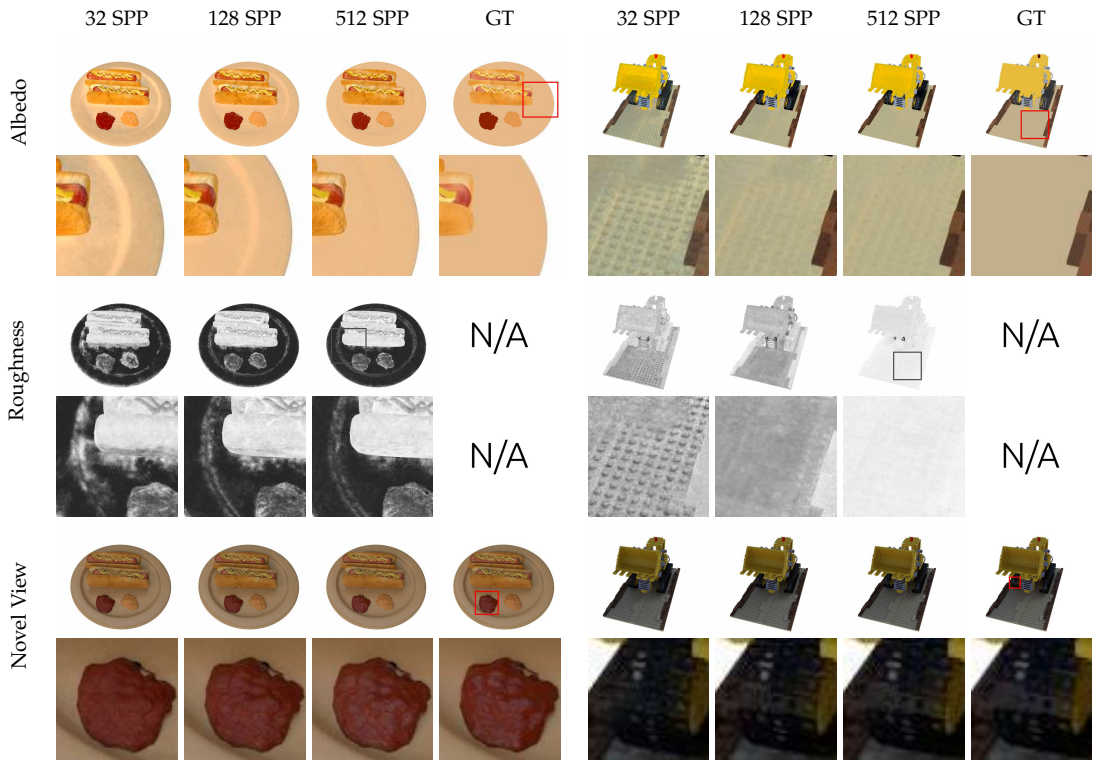

Figure 5.18: **Comparison on different SPP.** With an increase in the number of SPP in the training step, the albedo becomes more refined, roughness noise diminishes, and the novel view showcases enhanced specular effects. Note that the dataset lacks ground truth visualizations for roughness.

5.4.5 Multiple Importance Sampling

In this section, we investigate the impact of sampling strategies during training. For a fair comparison, all images are rendered with MIS in the test phase.

Surprisingly, Table 5.9 demonstrates that sampling strategies have a negligible impact on the performance of our model. Although the cos-weighted sampling strategy aids in better albedo reconstruction, the relighting performance remains consistent across all strategies, which indicates all strategies have a comparable quality in material property decomposition. Theoretically, MIS should reduce variance, thereby assisting the gradient descent optimizer to achieve better convergence. However, experiments show that this expectation does not hold in neural rendering. Our observation aligns with the findings of [HHM22]. They claim that the sampling strategies have only minimal impact when the scene predominantly features diffuse materials or those with low specularity, a condition that matches our scene. However, they found MIS to be particularly effective for high-specular materials, such as metals. We continue to incorporate MIS for our model, aiming to expand its potential capabilities.

Scene	Sampling Method	**Albedo**			**Novel View Synthesis**			**Relighting**		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hotdog	Random	30.881	0.947	0.032	38.981	0.975	0.014	31.906	0.941	0.036
	Cos-weighted	31.350	0.952	0.029	38.962	0.975	0.014	31.900	0.941	0.036
	BRDF	30.890	0.949	0.032	38.990	0.975	0.014	31.776	0.940	0.037
	MIS	30.639	0.947	0.032	38.983	0.975	0.014	31.788	0.940	0.037
Lego	Random	25.593	0.933	0.024	37.216	0.979	0.006	33.718	0.973	0.010
	Cos-weighted	26.060	0.937	0.024	37.313	0.979	0.005	33.802	0.973	0.010
	BRDF	25.437	0.931	0.024	37.358	0.979	0.005	33.717	0.973	0.010
	MIS	25.472	0.933	0.024	37.412	0.980	0.005	33.758	0.973	0.010

Table 5.9: **Quantitative comparison on sampling methods.** The choice of sampling method for training has minimal impact on the results.

6 Limitation and Future Work

Surface refinement We adopt a pre-optimized mesh as the 3D scene representation. The geometry is fixed during training and constrains our model performance. In future work, one can leverage the physically-based multi-bounce estimation to refine the geometry. As demonstrated in [Mun+22; HHM22], it is feasible to back-propagate the gradient directly to the mesh geometry representation. Alternatively, following the approach in [Jin+23; Mai+23], one can employ an implicit density field as the geometric representation and subsequently back-propagate to it for further refinement.

Advanced material modeling We adopt a basic BRDF model, where the diffuse and specular reflections are merely summed. This does not ensure energy conservation. Furthermore, our BRDF may not be expressive enough to model complicated light reflection. One can employ an advanced material model, such as (simplified) Disney BRDF [BS12]. Figure 6.1 illustrates the effect of the parameters for Disney BRDF. Another possible direction would be to explore the simulation of transmission, for instance, replace BRDF with BSDF, as exemplified in [Jak+22].

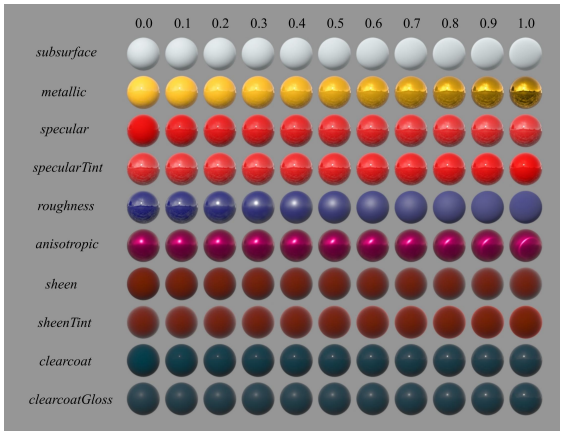

Figure 6.1: **Illustration of the effect of the parameters for Disney BRDF.** Image from [BS12].

Lighting assumption relaxing We assume that there is no light source in our scene. However, real-world settings frequently incorporate emitters or fluorescent materials. To address this issue, we need to relax our lighting assumption and include the reconstruction scene emitters. In indoor scene reconstruction tasks, numerous methods [Azi+19; Yu+23; Wu+23] have been devised to proficiently model these emitters. An example is shown in Figure 6.2. It is plausible that a fusion of the strengths of indoor scene emitter estimation and our environmental lighting estimation could offer a comprehensive solution.

Efficiency improvement Our model follows the classical path-tracing pipeline of computer graphics. This might not be optimal for inverse rendering. There could be opportunities to enhance its efficiency for neural rendering tasks without compromising its proficiency in global estimation [Wu+23]. Furthermore, our model demands a high SPP. Incorporating a denoiser [HHM22] has the potential to decrease SPP, leading to a faster training process.

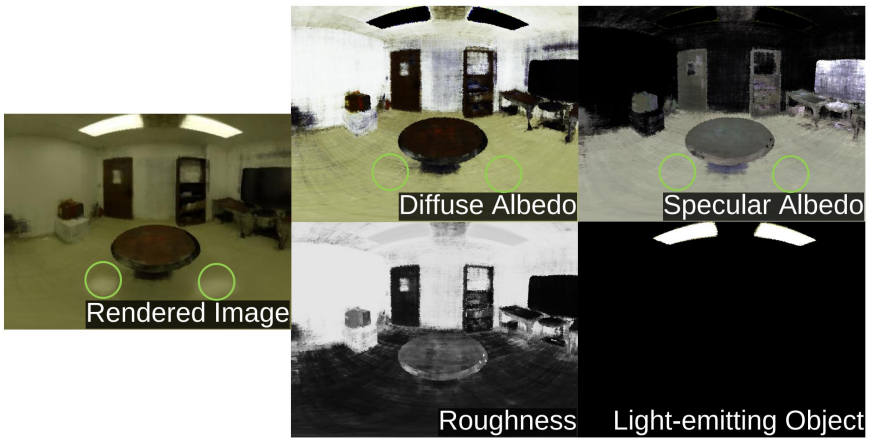

Figure 6.2: **Illustration of the indoor scene decomposition task.** In this task, the scene often contains emitters. The emitters of this scene are correctly reconstructed. Image from [Yu+23].

7 Conclusion

We introduce a neural rendering method proficient in global illumination estimation. Our method simulates real-world light transport by implementing a MC multi-bounce path tracer. We effectively back-propagate the gradients to material and lighting estimation. Without relying on handcrafted prior knowledge, our model adeptly reconstructs high-quality parameters for the spatial varying BRDF. Our model excels at precise shadowing and indirect illumination analyses. Furthermore, we perform an in-depth analysis examining the impact of multi-bounce estimation, delving into its advantages and associated costs. A comprehensive ablation study further measures the contribution of each component of our model. We demonstrate that our approach achieves comparable results with the state-of-the-art method, and clearly improves the reconstruction of secondary shading effects like self-reflection.

Abbreviations

BRDF Bidirectional Reflectance Distribution Function

MC Monte Carlo

SPP Samples Per Ray

MIS Multiple Importance Sampling

PBR Physically Based Rendering

GT Ground Truth

PDF Probability Density Function

SDF Signed Distance Function

BSDF Bidirectional Scattering Distribution Function

BTDF Bidirectional Transmittance Distribution Function

List of Figures

List of Tables

Bibliography

[Azi+19] D. Azinovic, T. Li, A. Kaplanyan, and M. Nießner. "Inverse Path Tracing for Joint Material and Lighting Estimation." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 2447–2456. DOI: 10.1109/CVPR.2019.00255.

[Bar+22] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 5460–5469. DOI: 10.1109/CVPR52688.2022.00539.

[Bar+23] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. "Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields." In: *ICCV* (2023).

[Bos+21] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. A. Lensch. "NeRD: Neural Reflectance Decomposition from Image Collections." In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 12664–12674. DOI: 10.1109/ICCV48922.2021.01245.

[BS12] B. Burley and W. D. A. Studios. "Physicallybased shading at disney." In: SIGGRAPH, volume 2012, pages 1–7., 2012.

[BS87] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. 1987.

[Fri+22] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. "Plenoxels: Radiance Fields without Neural Networks." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 5491–5500. DOI: 10.1109/CVPR52688.2022.00542.

[HHM22] J. Hasselgren, N. Hofmann, and J. Munkberg. "Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising." In: *NeurIPS*. 2022.

[Jak+22] W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang. *Mitsuba 3 renderer*. Version 3.0.1. https://mitsuba-renderer.org. 2022.

[Jen+14] R. R. Jensen, A. L. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. "Large Scale Multi-view Stereopsis Evaluation." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 406–413. DOI: 10.1109/CVPR.2014.59.

[Jin+23] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su. "TensoIR: Tensorial Inverse Rendering." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 165–174. DOI: 10.1109/CVPR52729.2023.00024.

[Kaj86] J. T. Kajiya. "The rendering equation." In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18-22, 1986*. Ed. by D. C. Evans and R. J. Athay. ACM, 1986, pp. 143–150. DOI: 10.1145/15922.15902.

[KB15] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." In: *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA, 2015.

[LC87] W. E. Lorensen and H. E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*. Ed. by M. C. Stone. ACM, 1987, pp. 163–169. DOI: 10.1145/37401.37422.

[Mai+23] A. Mai, D. Verbin, F. Kuester, and S. Fridovich-Keil. "Neural Microfacet Fields for Inverse Rendering." In: *CoRR* abs/2303.17806 (2023). DOI: 10.48550/arXiv.2303.17806. arXiv: 2303.17806.

[Mil+20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J. Frahm. Vol. 12346. Lecture Notes in Computer Science. Springer, 2020, pp. 405–421. DOI: 10.1007/978-3-030-58452-8_24.

[Mül21] T. Müller. *tiny-cuda-nn*. Version 1.7. Apr. 2021.

[Mun+22] J. Munkberg, W. Chen, J. Hasselgren, A. Evans, T. Shen, T. Müller, J. Gao, and S. Fidler. "Extracting Triangular 3D Models, Materials, and Lighting From Images." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 8270–8280. DOI: 10.1109/CVPR52688.2022.00810.

[Pho98] B. T. Phong. "Illumination for Computer Generated Pictures." In: *Seminal Graphics: Pioneering Efforts That Shaped the Field*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 95–101. ISBN: 158113052X.

[PJH16] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation: Third edition*. Nov. 2016, pp. 1–1233.

[Sch93] C. Schlick. "A Customizable Reflectance Model for Everyday Rendering." In: *Fourth Eurographics Workshop on Rendering*. Series EG 93 RW. Paris, France, 1993, pp. 73–84.

[Sri+21] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. "NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 7495–7504. DOI: 10.1109/CVPR46437.2021.00741.

[Tew+21] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, Z. Xu, T. Simon, M. Nießner, E. Tretschk, L. Liu, B. Mildenhall, P. P. Srinivasan, R. Pandey, S. Orts-Escolano, S. R. Fanello, M. Guo, G. Wetzstein, J. Zhu, C. Theobalt, M. Agrawala, D. B. Goldman, and M. Zollhöfer. "Advances in neural rendering." In: *SIGGRAPH 2021: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Courses, Virtual Event, USA, August 9-13, 2021*. ACM, 2021, 1:1–1:320. DOI: 10.1145/3450508.3464573.

[TR75] T. S. Trowbridge and K. P. Reitz. "Average irregularity representation of a rough surface for ray reflection." In: *J. Opt. Soc. Am.* 65.5 (1975), pp. 531–536. DOI: 10.1364/JOSA.65.000531.

[TS67] K. E. Torrance and E. M. Sparrow. "Theory for Off-Specular Reflection From Roughened Surfaces$*$." In: *J. Opt. Soc. Am.* 57.9 (1967), pp. 1105–1114. DOI: 10.1364/JOSA.57.001105.

[VG95] E. Veach and L. J. Guibas. "Optimally combining sampling techniques for Monte Carlo rendering." In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6-11, 1995*. Ed. by S. G. Mair and R. Cook. ACM, 1995, pp. 419–428.

[Wu+23] L. Wu, R. Zhu, M. B. Yaldiz, Y. Zhu, H. Cai, J. Matai, F. Porikli, T. Li, M. Chandraker, and R. Ramamoorthi. "Factorized Inverse Path Tracing for Efficient and Accurate Material-Lighting Estimation." In: *CoRR* abs/2304.05669 (2023). DOI: 10.48550/arXiv.2304.05669. arXiv: 2304.05669.

[Yu+22] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger. "MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction." In: *NeurIPS*. 2022.

[Yu+23] B. Yu, S. Yang, X. Cui, S. Dong, B. Chen, and B. Shi. "MILO: Multi-Bounce Inverse Rendering for Indoor Scene With Light-Emitting Objects." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.8 (2023), pp. 10129–10142. DOI: 10.1109/TPAMI.2023.3244658.

[Zel+21] T. Zeltner, S. Speierer, I. Georgiev, and W. Jakob. "Monte Carlo estimators for differential light transport." In: *ACM Trans. Graph.* 40.4 (2021), 78:1–78:16. DOI: 10.1145/3450626.3459807.

[Zha+18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068.

[Zha+21a] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely. "PhySG: Inverse Rendering With Spherical Gaussians for Physics-Based Material Editing and Relighting." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 5453–5462. DOI: 10.1109/CVPR46437.2021.00541.

[Zha+21b] X. Zhang, P. P. Srinivasan, B. Deng, P. E. Debevec, W. T. Freeman, and J. T. Barron. "NeRFactor: neural factorization of shape and reflectance under an unknown illumination." In: *ACM Trans. Graph.* 40.6 (2021), 237:1–237:18. DOI: 10.1145/3478513.3480496.

[Zha+22] Y. Zhang, J. Sun, X. He, H. Fu, R. Jia, and X. Zhou. "Modeling Indirect Illumination for Inverse Rendering." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 18622–18631. DOI: 10.1109/CVPR52688.2022.01809.