Supplementary Material for NeFII: Inverse Rendering for Reflectance Decomposition with Near-Field Indirect Illumination

A. Sampling from SG Sampling Function

As described in Sec. 3.3, we utilize the Spherical Gaussian (SG) distribution sampling method to improve the Monte Carlo ray sampling efficiency:

$$p_{SG}(\boldsymbol{w}_i) = \sum_{k=1}^{M} a_k \frac{\lambda_k}{2\pi(1-e^{-2\lambda_k})} e^{\lambda_k(\boldsymbol{w}_i \cdot \boldsymbol{\xi}_k - 1)}, \tag{1}$$

where $\boldsymbol{\xi}, \lambda, \boldsymbol{\mu} \in \boldsymbol{\Theta}_E$ are SG parameters of environment illumination, *i.e.*, lobe axis, lobe sharpness and lobe amplitude of SG respectively.

When sampling, we first utilize the probability a_k to decide which Gaussian component to draw from, then draw $\boldsymbol{w}_i$ from the k-th Gaussian distribution. In order to draw samples from the k-th Gaussian distribution, we apply inverse transform sampling [3], which employs uniform random variables and maps them to random variables of the target distribution. We first transform the PDF of direction $\boldsymbol{w}_i$ to 1D marginal and conditional density functions of its spherical coordinate φ and θ. Following [3], the joint PDF $p_k(\theta, \varphi)$ of spherical coordinate φ and θ is derived as

$$p_k(\theta, \varphi) = c_k \sin\theta e^{\lambda_k(\cos\theta - 1)}. \tag{2}$$

Hence, the marginal density function $p_k(\theta)$ of θ is

$$\begin{aligned} p_k(\theta) &= \int_0^{2\pi} c_k \sin\theta e^{\lambda_k(\cos\theta-1)} d\varphi \\ &= 2\pi c_k \sin\theta e^{\lambda_k(\cos\theta-1)}. \end{aligned} \tag{3}$$

As a result, the conditional density function $p_k(\varphi|\theta)$ of φ is

$$p_k(\varphi|\theta) = \frac{p_k(\theta, \varphi)}{p_k(\theta)} = \frac{1}{2\pi}. \tag{4}$$

We then compute the cumulative distribution function (CDF) of the distribution, $P_k(\theta)$ and $P_k(\varphi|\theta)$:

$$\begin{aligned} P_k(\theta) &= \int_0^{\theta} 2\pi c_k \sin t e^{\lambda_k(\cos t-1)} dt \\ &= \frac{2\pi c_k}{\lambda_k}(1 - e^{\lambda_k(\cos\theta-1)}), \end{aligned} \tag{5}$$

Figure 1. Comparison of albedo results of real scenes.

$$P_k(\varphi|\theta) = \int_0^{\varphi} \frac{1}{2\pi} = \frac{\varphi}{2\pi}. \tag{6}$$

According to inverse transform sampling, random variable $X = F_X^{-1}(u)$ has distribution $F_X(x)$, where u is a random value generated from the standard uniform distribution. Hence, to apply inverse transform sampling and draw a sample θ based on a uniformly distributed random number u_1, we solve for $P_k(\theta) = u_1$:

$$\begin{aligned} &\frac{2\pi c_k}{\lambda_k}(1 - e^{\lambda_k(\cos\theta-1)}) = u_1 \\ \Rightarrow \theta = &\arccos(1 + \frac{1}{\lambda_k}\ln(1 - \frac{\lambda_k u_1}{2\pi c_k})). \end{aligned} \tag{7}$$

In a similar way, we can draw a sample φ based on a uniformly distributed random value u_2 as:

$$\varphi = 2\pi u_2. \tag{8}$$

B. Comparison of albedo results of real scenes.

Fig. 1 illustrates the comparison between our method and Invrender in recovering diffuse materials in real scenes. Our method outperforms Invrender in modeling indirect illumination, which helps to avoid baking indirect illumination into the albedo and causing incorrect brightness of certain areas.

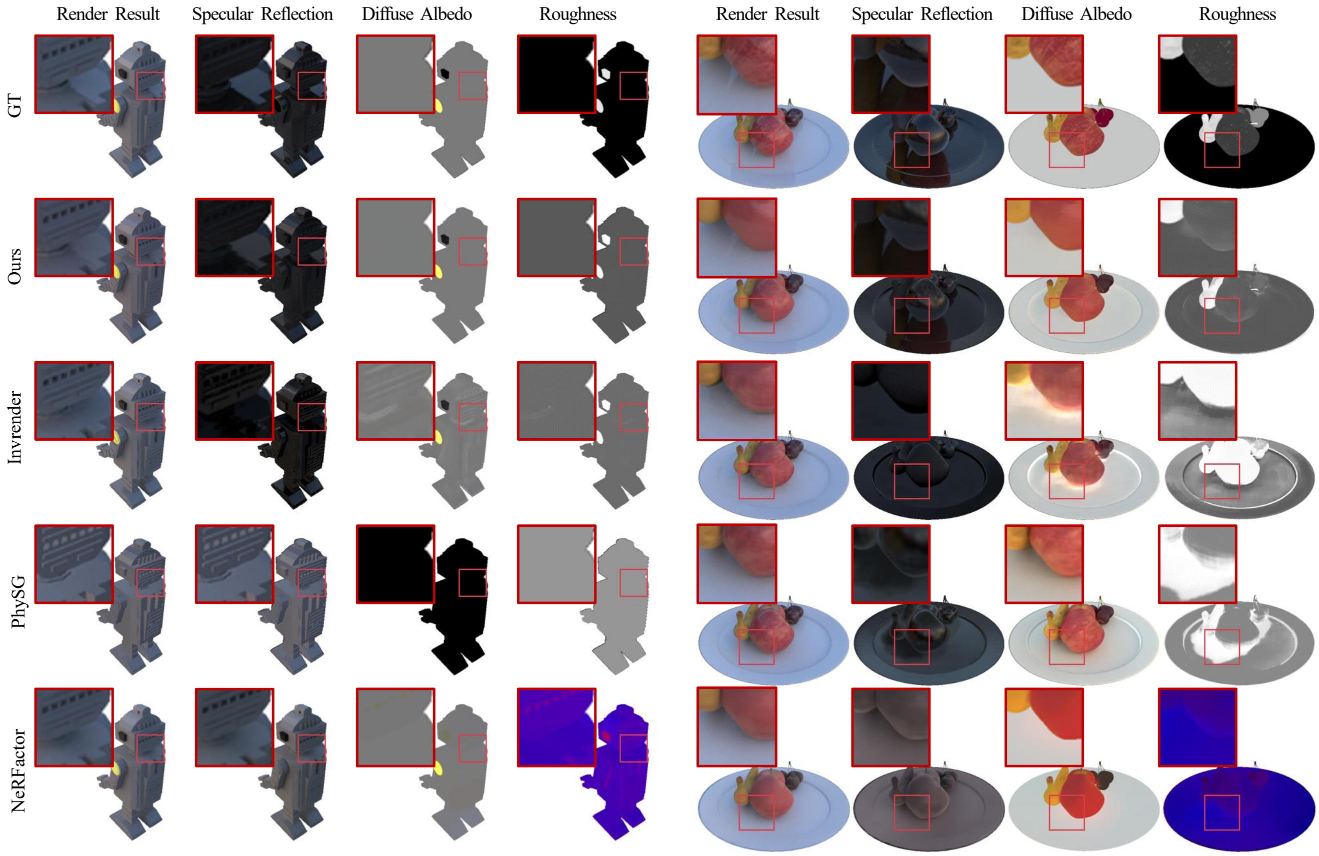

Figure 2. Additional results of synthetic scenes.

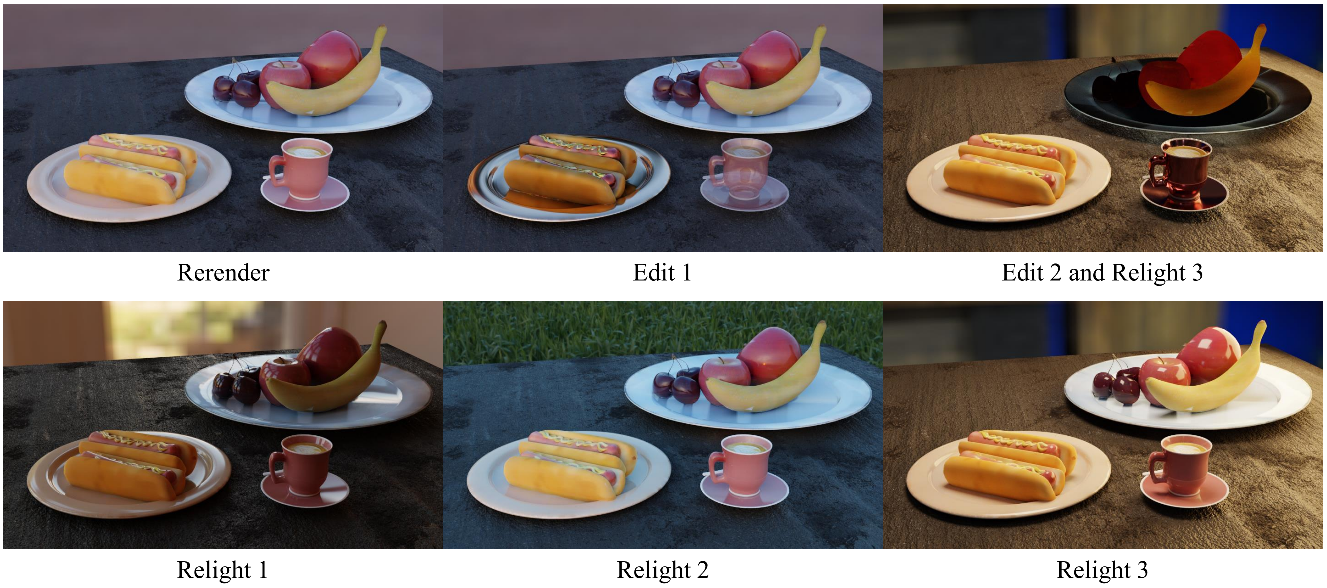

Figure 3. **Results of Scene Manipulation in Blender.** We present material editing and relighting results in Blender [2] of our recovered models, *i.e*., hotdog, coffee, and fruits. The results are rendered by Blender Cycles at 2048 ssp.

C. Additional Results of synthetic scenes.

Fig. 2 shows qualitative comparisons results of the other synthetic scenes.

D. Scene Manipulation in Blender

Our method supports further scene manipulation in graphics engines. We convert our recovered material properties to image textures and import them into Blender [2] based on OpenMVS [1]. In Fig. 3, we present the material editing and relighting results of our recovered models, hotdog, coffee, and fruits. Note that there are some biases of image textures caused by OpenMVS.

References

[1] Dan Cernea. OpenMVS: Multi-view stereo reconstruction library. 2020. 3

[2] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 2, 3

[3] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 1