Modeling Indirect Illumination for Inverse Rendering

Yuanqing Zhang[1,2] Jiaming Sun[1] Xingyi He[1] Huan Fu[2] Rongfei Jia[2] Xiaowei Zhou[1*]
[1]Zhejiang University [2]Tao Technology Department, Alibaba Group

Abstract

Recent advances in implicit neural representations and differentiable rendering make it possible to simultaneously recover the geometry and materials of an object from multi-view RGB images captured under unknown static illumination. Despite the promising results achieved, indirect illumination is rarely modeled in previous methods, as it requires expensive recursive path tracing which makes the inverse rendering computationally intractable. In this paper, we propose a novel approach to efficiently recovering spatially-varying indirect illumination. The key insight is that indirect illumination can be conveniently derived from the neural radiance field learned from input images instead of being estimated jointly with direct illumination and materials. By properly modeling the indirect illumination and visibility of direct illumination, interreflection- and shadow-free albedo can be recovered. The experiments on both synthetic and real data demonstrate the superior performance of our approach compared to previous work and its capability to synthesize realistic renderings under novel viewpoints and illumination. Our code and data are available at https://zju3dv.github.io/invrender/.

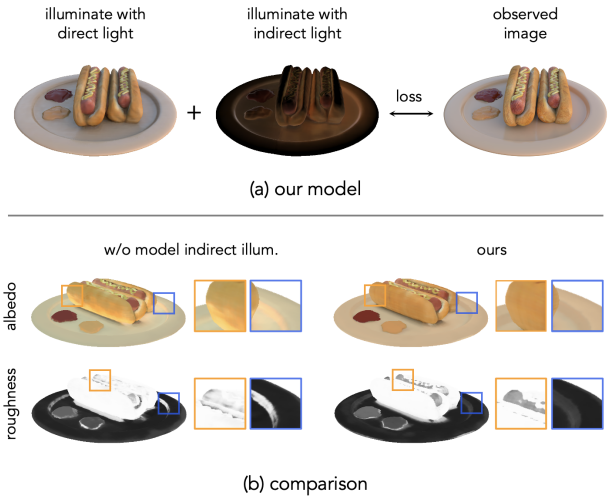


Figure 1. To precisely recover SVBRDF (parameterized as albedo and roughness) from multi-view RGB images, we propose an efficient approach to reconstruct spatially varying indirect illumination and combine it with environmental light evaluated by visibility as the full light model (a). The example in (b) demonstrates that without modeling indirect illumination, its rendering effects are baked into the estimated albedo to compensate for the incomplete light model and also result in artifacts in the estimated roughness.

1. Introduction

Recovering the geometry, materials, and lighting of a 3D scene from images, also known as inverse rendering, has been a long-standing problem in the fields of computer vision and graphics. It is gaining traction in this era of blowout VR and AR applications, where there is a high demand for easily acquired 3D contents from the real world. Previous capture systems, such as light-stages with controlled light directions and cameras [8, 11, 31], using a co-located flashlight and camera in a dark room [2, 3], and rotating objects with a turntable [7, 26], show limitations in user-friendliness.

More recent works [5,29,32] explore flexible capture settings under natural illumination. These methods typically represent geometry and spatially varying BRDF (SVBRDF) as coordinate-based neural networks and recover them by optimizing a re-rendering loss that compares rendered images with input images. However, capturing under natural illumination often shows complex effects such as soft shadows and interreflections. It is intractable to simulate these effects when optimizing SVBRDF and light parameters as it necessitates expensive recursive path tracing in physically based rendering. Prior methods usually ignore both self-occlusion and interreflection [29] in order to reduce computation, or only model visibility [32] or limit the indirect lighting to a single bounce with known light sources [22]. Without properly modeling the indirect illumination, there exists a gap between the captured image and the rendered image. As a result, the effect of indirect illumination in the captured images is prone to being baked into the estimated diffuse albedo to compensate for this gap, as illustrated in

The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG. This work was done when Yuanqing Zhang was an intern at Alibaba Group. *Corresponding author: Xiaowei Zhou.

Figure 1. It also results in artifacts in the recovered specular reflectance and environmental light as they explain the observed images together with albedo.

In this paper, we aim to estimate the SVBRDF of objects from multi-view RGB images captured under unknown static illumination. Our main technical innovation is an efficient approach to modeling indirect illumination in this inverse rendering process. We model the indirect illumination by a multilayer perceptron (MLP) that maps a 3D surface point to its indirect incoming illumination. The core idea to efficiently learning this indirect illumination MLP is that the indirect illumination doesn't need to be jointly learned with the SVBRDF and environmental light, but can be directly derived from the outgoing radiance field of the scene, which can be constructed from multi-view images with the off-the-shelf neural scene representation methods (e.g., [15, 27]).

Specifically, we first learn the geometry and outgoing radiance field of the object, both represented as MLPs, from the input images using the existing method [27]. Then, the learned radiance field serves as the ground-truth incoming illumination of its reachable surface points to train the indirect illumination MLP. Finally, the learned indirect illumination is plugged into the rendering equation and fixed during the optimization of SVBRDF and environmental light. In this way, the indirect illumination can be directly queried when optimizing the other unknowns without the need of recursive path tracing, making the inverse rendering problem better constrained and more efficient to solve. Furthermore, to reduce the ambiguity of disentangling BRDF and incident light, we introduce a prior that a real-world object should consist of limited types of materials. This prior is imposed by representing SVBRDF as an encoder-decoder with a sparse latent space.

We evaluate the proposed method on both synthetic and real datasets. The experimental results show that our approach outperforms baseline methods and is able to recover shadow- and interreflection-free albedo and high-quality roughness, as well as supporting realistic free-viewpoint relighting.

2. Background

Inverse rendering. Inverse rendering, the task of decomposing the image appearance into the underlying intrinsic properties such as geometry, material, and lighting conditions, has been a longstanding problem in computer vision and graphics. The full inverse rendering problem in its most general form is well-known to be severely ill-posed. The key problem in inverse rendering is to properly add priors and regularizations to the optimization process to mitigate the ill-posed condition.

Single-image inverse rendering methods [1, 12–14, 18, 21, 25, 28] rely heavily on the strong prior of the planar geometry. Since the planar input and output maps are naturally easier to be processed by CNNs, these methods can learn priors for normal, reflectance, and illumination from large-scale datasets. They can effectively infer plausible materials and normal maps from a single image but usually cannot recover spatially-varying 3D representations of these factors.

Most methods that recover fully factorized 3D geometry, materials and lighting require scenes to be captured under more constrained settings. They either capture images while rotating the object with the camera fixed [7, 26], or shoot a video using a handheld cellphone with a flash in a dark environment so that the point light is associated with the camera and its location is known [2–4, 16, 19]. The varied or known illumination provide rich information for inferring geometry and material properties.

Implicit neural representation. Recent advances in implicit neural representation enable new possibilities for inverse rendering. NeRF [15] achieves photo-realistic novel view synthesis by representing radiance fields with multilayer perceptrons. Supervised with differentiable volumetric rendering, NeRF is able to reconstruct the radiance field of a scene with only a collection of images. While NeRF represents geometry as volumetric density fields, some surface-based methods like IDR [27] and NeuS [24] represent geometry with Signed Distance Functions (SDFs). These methods work well for novel view synthesis, but they only model the outgoing radiance of a surface and are not capable of disentangling it into the incoming radiance and the underlying material property. As a result, they don't enable free-viewpoint relighting.

Inverse rendering with implicit neural representation. Enabled by the fully differentiable pipelines in implicit neural representation, recent methods in inverse rendering aim at more "casual" capture conditions. Notably, PhySG [29] and NeRFactor [32] decompose the scene under complex and unknown illumination. NeRD [5] extends NeRF to deal with captures under fixed or varying illumination. However, all these methods only consider direct illumination from the light source and ignore indirect illumination, so they are unable to simulate interreflection effects to resemble the observed images. As a result, they can only model simple and convex surfaces that have neglectable indirect light. NeRV [22] does consider one indirect bounce, but with known environment light and is rendered with Monte-Carlo ray tracing. Our method is able to reconstruct high quality indirect light with unconstrained bounces and does not require known lighting conditions.

The rendering equation. For non-emitted object, the rendering equation computes the outgoing radiance L_o at

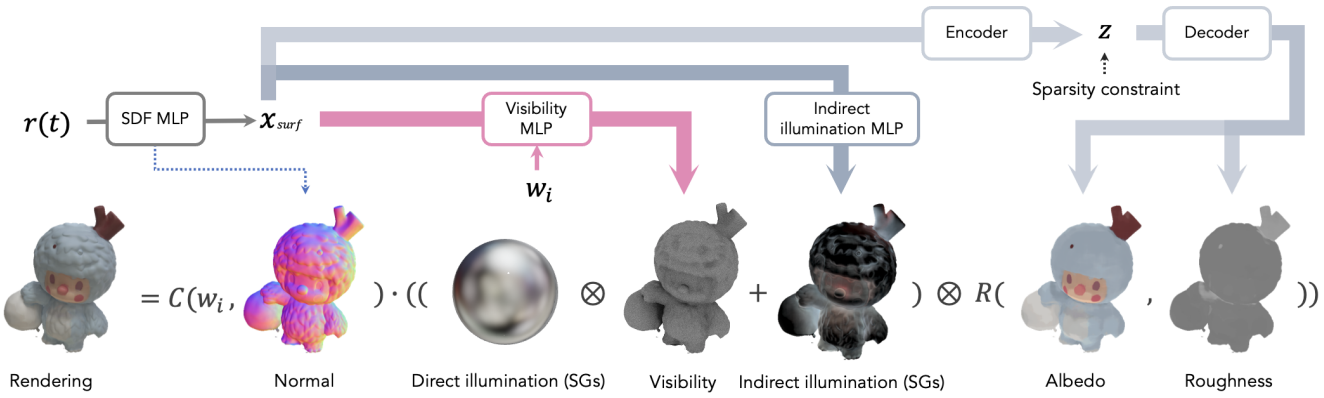


Figure 2. **Forward rendering.** For a specific surface point $\hat{\mathbf{x}}$, the full incoming light is modeled as the sum of direct illumination direction-wise multiplied by visibility and indirect illumination derived from the reconstructed outgoing radiance field. The spatially varying BRDF parameters are output from an encoder-decoder network with a sparsity constraint on the latent code, and each specular BRDF is further transformed to a single spherical Gaussian (SG). During forward rendering, only the BRDF and the direct illumination need to be optimized, while the others are all pre-acquired and fixed. In the bottom row, the visualized visibility is the mean value over all directions and the indirect illumination is the irradiance at each point.

surface point $\hat{\mathbf{x}}$ along direction $\boldsymbol{\omega}_o$ by integrating the reflected light over hemisphere [9]:

$$L_o(\hat{\mathbf{x}}, \boldsymbol{\omega}_o) = \int_\Omega L_{in}(\hat{\mathbf{x}}, \boldsymbol{\omega}_i) f_r(\hat{\mathbf{x}}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)(\boldsymbol{\omega}_i \cdot \mathbf{n}) \mathrm{d}\boldsymbol{\omega}_i \quad (1)$$

The $L_{in}(\hat{\mathbf{x}}, \boldsymbol{\omega}_i)$ is the incoming radiance at surface point $\hat{\mathbf{x}}$ along direction $\boldsymbol{\omega}_i$ and the BRDF function f_r describes how much light arriving from direction $\boldsymbol{\omega}_i$ is reflected towards direction $\boldsymbol{\omega}_o$ at $\hat{\mathbf{x}}$.

3. Method

3.1. Overview

Given a set of posed images of an object captured under static illumination, we learn to decompose the shape and SVBRDF to enable applications such as free-view relighting. We solve the inverse rendering problem in an analysis-by-synthesis manner, where we optimize the parameters of the forward rendering model until the rendered images closely resemble the observed images. Figure 2 depicts the forward rendering process of our proposed method.

In the paper, we represent the geometry as a zero level set as IDR [27] by learning a Signal Distance Function (SDF), parameterized by a multilayer perceptron $S(\mathbf{x})$, that maps from a 3D location $\mathbf{x}$ to the SDF value at this location. It gives smooth and realistic surfaces of objects. We decompose the spatially varying incoming light $L_{in}(\hat{\mathbf{x}}, \boldsymbol{\omega}_i)$ at a surface point $\hat{\mathbf{x}}$ along the direction $\boldsymbol{\omega}_i$ into two components: direct illumination E evaluated by visibility (Sec. 3.2) and indirect illumination L_i efficiently derived from the outgoing radiance field (Sec. 3.3). In contrast to previous works, the SVBRDF parameters in our formulation are parameterized as an encoder-decoder network with a sparse latent space (Sec. 3.4).

To render a camera ray, the intersection $\hat{\mathbf{x}}$ of the ray and SDF surface can be observed via the sphere tracing technique, and its corresponding surface normal is the gradient of the SDF: $\mathbf{n} = \nabla_{\hat{\mathbf{x}}} S$. Then we query visibility, indirect illumination, diffuse albedo and roughness from networks, and perform rendering together with environment lighting (Sec. 3.5). The parameters of SVBRDF and direct illumination are optimized by minimizing the reconstruction error between the renderings and the observed images.

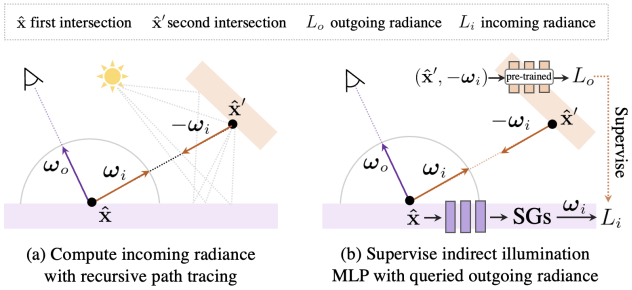
3.2. Visibility for Direct Illumination

For direct illumination, we assume that all lights come from an infinitely faraway environment and parameterize them as M=128 spherical Gaussians (SGs) [23]:

$$E(\boldsymbol{\omega_i}) = \sum_{k=1}^{M} G(\boldsymbol{\omega_i}; \boldsymbol{\xi}_k, \lambda_k, \boldsymbol{\mu}_k) \quad (2)$$

where $\boldsymbol{\xi} \in \mathbb{S}^2$ is the lobe axis, $\lambda \in \mathbb{R}_+$ is the lobe sharpness, and $\boldsymbol{\mu} \in \mathbb{R}^3$ is the lobe amplitude.

The environment lighting is evaluated by the visibility indicating whether the direction $\boldsymbol{\omega}_i$ at surface point $\mathbf{x}$ is occluded or not. The visibility can be obtained by performing sphere tracing from surface points to light sources. However, the tracing step is repeatedly executed during forward rendering and is time-consuming. So we re-parameterize it as an MLP that maps the surface point location $\mathbf{x}$ and direction $\boldsymbol{\omega}_i$ to visibility: $V(\mathbf{x}, \boldsymbol{\omega}_i) \mapsto v$. The network provides a compact and continuous representation and requires only a small number of sampled rays above surface points for training. The direction-wise multiplication of the visibility



(a) Compute incoming radiance with recursive path tracing

(b) Supervise indirect illumination MLP with queried outgoing radiance

Figure 3. Instead of computing incoming radiance by performing costly recursive path tracing (a), we consider the pre-trained outgoing radiance field as indirect illumination and train a network that maps a 3D location to its indirect incoming illumination represented as a mixture of SGs (b).

function and the environment lighting SG is desired to yield another SG to support the integral of the spherical functions during rendering. We achieve this by having the amplitude of the output SG produce the same integrated value as the original lobe and preserving its center:

$$V(\mathbf{x}, \boldsymbol{\omega_i}) \otimes G(\boldsymbol{\omega_i}; \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \approx G(\boldsymbol{\omega_i}; \boldsymbol{\xi}, \boldsymbol{\lambda}, \gamma\boldsymbol{\mu}) \quad (3)$$

$$\gamma = \frac{\sum_{k=1}^{S} G(\boldsymbol{\omega}_k) V(\mathbf{x}, \boldsymbol{\omega}_k)}{\sum_{k=1}^{S} G(\boldsymbol{\omega}_k)} \quad (4)$$

The visibility ratio γ is obtained by randomly sampling the $S = 32$ directions in the SG lobe and taken a weighted average of queried visibility.

3.3. Indirect Illumination

According to the rendering equation, the indirect incoming radiance $L_i(\hat{\mathbf{x}}, \boldsymbol{\omega}_i)$ at the intersection $\hat{\mathbf{x}}$ of the camera ray and surface toward direction $\boldsymbol{\omega}_i$ is obtained by first performing ray tracing, and then assigned by the outgoing radiance $L_o(\hat{\mathbf{x}}', -\boldsymbol{\omega}_i)$ of the second intersection $\hat{\mathbf{x}}'$ toward direction $-\boldsymbol{\omega}_i$:

$$L_i(\hat{\mathbf{x}}, \boldsymbol{\omega}_i) = L_o(\hat{\mathbf{x}}', -\boldsymbol{\omega}_i) \quad (5)$$

$L_o(\hat{\mathbf{x}}', -\boldsymbol{\omega}_i)$ is rendered by continuing sampling and integrating rays over the hemisphere, as illustrated in Figure 3. As the number of considered bounces increases, the tracing and rendering computation grows with the *exponential* order of the sample amount. It is typically intractable in reality and increases the complexity of decomposing unknowns from rendering.

We tackle this problem by reconstructing the outgoing radiance field and deriving indirect illumination from it, rather than performing exhaustive ray tracing for the indirect illumination. The outgoing radiance field, which can be viewed as a neural renderer, is a continuous function of the surface point location $\hat{\mathbf{x}}$, normal $\hat{\mathbf{n}}$ and viewing direction $\boldsymbol{\omega_o}$: $R(\hat{\mathbf{x}}, \hat{\mathbf{n}}, \boldsymbol{\omega_o}) \mapsto L_o$. We learn this field parameterized as an MLP from observed images together with geometry using view synthesis method [27]. Therefore, the outgoing radiance of the second intersection, which is the cumulative results of multiple bounces, is obtained by querying the MLP:

$$L_o(\hat{\mathbf{x}}', -\boldsymbol{\omega}_i) = R(\hat{\mathbf{x}}', \hat{\mathbf{n}}', -\boldsymbol{\omega}_i) \quad (6)$$

where $\hat{\mathbf{n}}'$ is the normal of the second intersection.

We further transfer it into indirect illumination represented as a mixture of SGs and cache it in an MLP to avoid duplicate computation of tracing from $\hat{\mathbf{x}}$ to $\hat{\mathbf{x}}'$. The representation facilitates the hemispherical integration with other SG lobes, thus avoiding the use of the Monte-Carlo method, which requires a trade-off between low-cost sampling and high-quality rendering. Here, we introduce the indirect illumination MLP $I(\mathbf{x})$ that outputs the SG parameters $\Gamma \in \mathbb{R}^{24\times7}$ at any input 3D location $\mathbf{x}$. The incoming radiance is determined by querying the SG function at the desired surface point and direction:

$$L_i(\hat{\mathbf{x}}, \boldsymbol{\omega}_i) = G(\boldsymbol{\omega}_i; I(\hat{\mathbf{x}})) \quad (7)$$

The indirect illumination MLP is supervised by first drawing samples from outgoing radiance field R, and then forcing the incoming radiance to reproduce the corresponding outgoing radiance. We visualize this process in Figure 3.

3.4. BRDF

We use the simplified Disney BRDF model [6] with diffuse albedo $\mathbf{a}$ and roughness $\mathbf{r}$ as parameters and assume dielectric materials with fixed $F_0 = 0.02$ in the Fresnel term.

Parameterizing SVBRDF by directly mapping surface points to its parameters is straightforward. However, it often leads to noisy roughness since a few surface points lack supervision due to the distribution of the training views or self-occlusion. We alleviate this problem by introducing a prior that an object is usually composed of a small amount of materials.

Our solution is to represent SVBRDF as an encoder-decoder network with a sparse latent space. The network transforms the input surface point $\mathbf{x}$ to its corresponding latent code $\mathbf{z}$ and decodes it to its diffuse albedo and roughness. We impose a sparsity constraint [17] on the latent code so that most of the channels in $\mathbf{z}$ are close to zero:

$$\ell_{\mathrm{KL}} = \sum_{j=1}^{n} \mathrm{KL}(\rho \parallel \hat{\rho}_j) \quad (8)$$

where $\mathrm{KL}(\rho \parallel \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$ is a Kullback-Leibler divergence loss and $\hat{\rho}_j$ is the average of j^{th} channel of $\mathbf{z}$ over batch input. ρ is set to 0.05. n is the length of latent code. We further apply a smooth loss on the

decoder D such that close latent codes are clustered to yield same SVBRDF:

$$\ell_{\mathrm{s}} = ||D(\mathbf{z}) - D(\mathbf{z} + \boldsymbol{\xi})||_1 \tag{9}$$

where $\boldsymbol{\xi}$ is a small random variable drawn from a normal distribution with zero mean and 0.01 variance.

3.5. Rendering

The BRDF function f_r in Equation 1 contains a diffuse component $\frac{\mathbf{a}}{\pi}$ and a specular component $f_s(\hat{\mathbf{x}}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$. We convert both the specular BRDF f_s and the clamped cosine factor $C = \boldsymbol{\omega}_i \cdot \mathbf{n}$ to a single SG as in prior work [29]. So Equation 1 can be approximated as the fast inner product of SGs. Specifically, we separate the rendering of direct illumination into diffuse component L_d and specular component L_s. The diffuse component is calculated as the sum of the integrals of each masked environment lighting SG and the clamped cosine factor:

$$L_d(\hat{\mathbf{x}}) = \frac{\mathbf{a}}{\pi} \sum_{k=1}^{M} (V(\hat{\mathbf{x}}, \boldsymbol{\omega}_i) \otimes E_k(\boldsymbol{\omega}_i)) \cdot C \tag{10}$$

Note that in the specular component, in order to accurately approximate the final integral in the presence of a narrow specular lobe, the visibility ratio γ is determined by sampling the specular SG:

$$L_s(\hat{\mathbf{x}}, \boldsymbol{\omega}_o) = \sum_{k=1}^{M} (f_s \otimes V(\hat{\mathbf{x}}, \boldsymbol{\omega}_i)) \otimes E_k(\boldsymbol{\omega}_i) \cdot C \tag{11}$$

As for the rendering of indirect illumination, the spatially varying indirect illumination is first queried from the indirect illumination MLP I, and the rendering is similar to the above process, except that the visibility is not required.

3.6. Training

We optimize the geometry, SVBRDF and environment lighting from a set of posed images through three-stage training. First, the SDF MLP $S(\mathbf{x})$ and outgoing radiance MLP R are optimized using [29]. Second, we sample 256 surface points and draw 16 sampled rays for each, then perform sphere tracing to obtain visibility and incoming radiance simultaneously, which serve as the ground truth for supervising the visibility MLP V and indirect illumination MLP I via cross-entropy loss and ℓ_1 loss. Last, the diffuse albedo, roughness and direct illumination are jointly optimized by minimizing the reconstruction loss ℓ_{recon} between the renderings and the observed images. The full loss in the final stage is:

$$\ell = \lambda_{\mathrm{recon}} \ell_{\mathrm{recon}} + \lambda_{\mathrm{KL}} \ell_{\mathrm{KL}} + \lambda_{\mathrm{s}} \ell_{\mathrm{s}} \tag{12}$$

We set weights $\lambda_{\mathrm{recon}} = 1.0$, $\lambda_{\mathrm{KL}} = 0.01$, $\lambda_{\mathrm{s}} = 0.1$ in our experiments.

The architecture of visibility MLP, indirect illumination MLP and encoder of BRDF contains 4 layers with 512 hidden units. Positional encoding [2] is applied to the input 3D locations and directions with 10 and 4 frequency components, respectively. The decoder of BRDF is a 2-layer network with a 32-dimensional input latent code and 128 hidden units. We implement our model in PyTorch and optimize using Adam [10] with learning rate $5e^{-4}$. Both of the latter two stages run 200 epochs on a single RTX 3090 GPU, which takes about 1 and 2 hours, respectively.

4. Experiments

In this section, we conduct experiments to investigate the performance of our inverse rendering approach. First, we briefly present how we build a synthetic dataset to examine our setting in Sec. 4.1. Then, we make quantitative comparisons with two baselines on the synthetic data in Sec. 4.2. Third, we perform several ablations to discuss our key components in Sec. 4.3. Finally, we qualitatively study the inverse rendering and relighting abilities of our method on the real dataset in Sec. 4.4. We refer to the supplemental materials for more results.

4.1. Synthetic Data

We collect 4 CAD models, each with obvious self-occlusions and multiple materials. For a specific object, we assign it with a natural environment map, and render 100 training images as well as their masks via Blender Cycles. Masks are required by the SDF learning process [27]. We render other 200 test images as well as their albedo and roughness maps to evaluate the novel view synthesis performance and the inverse rendering ability. To measure the relighting performance, we utilize other two environment maps and render 200 images for each case. The image resolution is 800×800.

4.2. Baseline Comparisons

To our best knowledge, there are only a few works that study the exactly same inverse rendering setting as this paper, *i.e,* training with fixed unknown illumination while supporting free-view relighting. We take NeRFactor [32] and PhySG [29] as baselines and make quantitative comparisons on the synthetic datasets. The image quality metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [30]. Since there is an inevitable scale ambiguity in estimating the albedo and environment lighting, we additionally evaluate the albedo after aligning with the ground truth, as done in [29, 32].

NeRFactor distills the volumetric geometry of NeRF [15] into a surface representation. It relies on a data-driven BRDF prior learned from real-world BRDF measurements

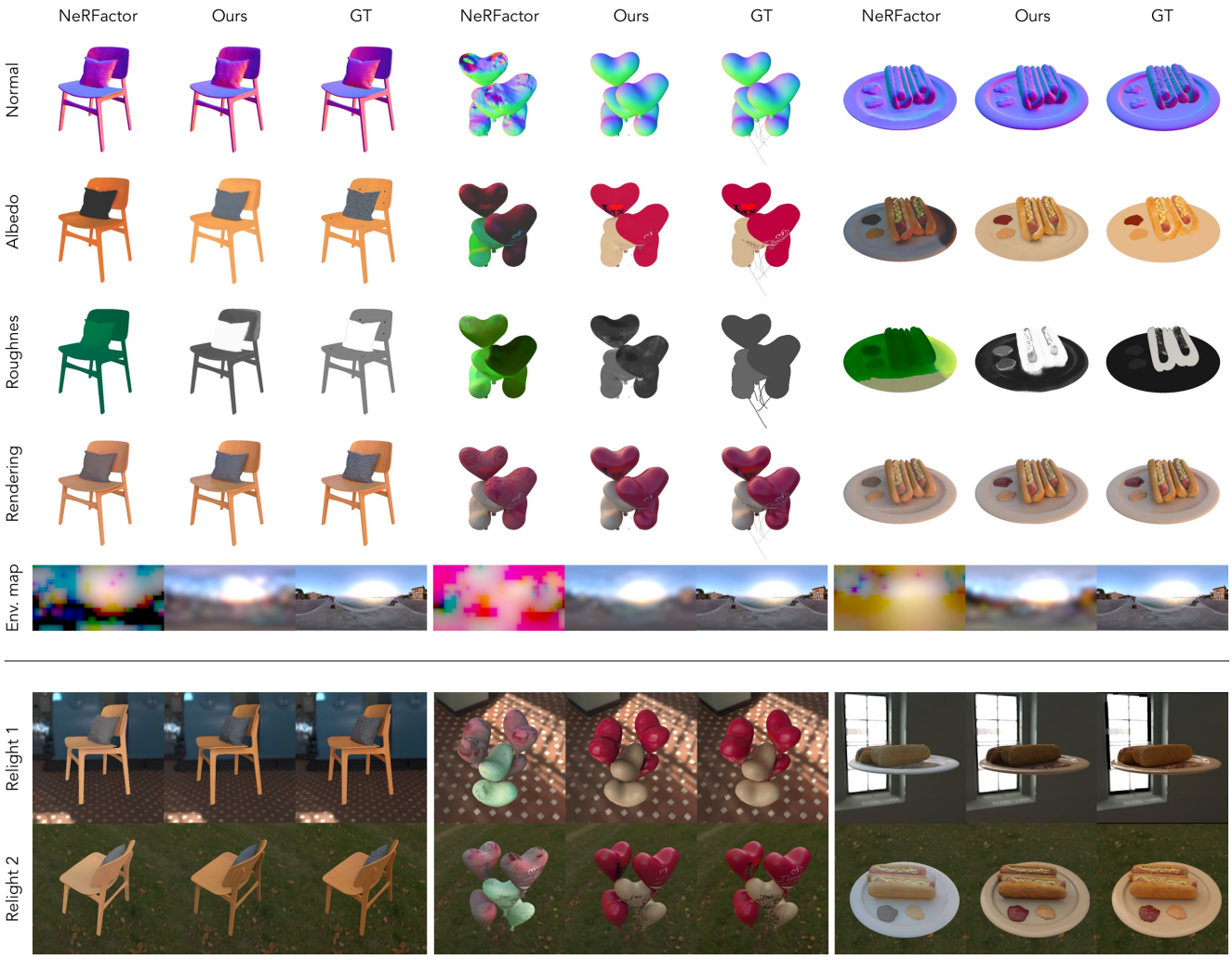


Figure 4. **Comparisons with previous work.** We visualize the estimated normal, diffuse albedo, roughness and environment map of NeRFactor [32] and our method on three scenes. Note that the roughness of NeRFactor is visualized with the latent code, which represents a BRDF identity, since it is parameterized by a learned model. We also compare the re-renderings under a novel view and original light (the fourth row) as well as novel views and novel light (the last two rows).

to recover 3D neural fields of SVBRDF. Table 1 and Figure 4 demonstrate that our method is superior to NeRFactor both quantitatively and qualitatively. NeRFactor parameterizes illumination as a 16×32 resolution environment map so that each pixel/parameter can vary independently. The estimated results show that the albedo would easily be baked into the environment map during its optimization process, thus resulting in poor relighting performance. In contrast, our predicted environment lighting and SVBRDF contain fine details and are visually close to the ground truth maps, as shown in Figure 4.

PhySG is able to jointly recover environmental lighting, BRDFs and geometry from multi-view inputs captured under static illumination. However, it presumes that the recovered object is homogeneous. We adapt its pipeline by replacing its global roughness with a spatially varying roughness parameterized as an MLP. The experimental results in Table 1 show that it performs badly. The main reason is that, without modeling visibility and indirect illumination, geometry optimization is highly ill-posed, especially in areas with obvious shadow and interreflection.

4.3. Ablation Studies

We ablate combinations of three components of our methods that primarily affect the inverse rendering quality. We argue that a slight improvement over the studied metrics may bring an upgraded visual experience as rendering is a detailed effect. The results are reported in Table 1 and Figure 5.

In "w/o vis. & ind. illum.", we train a model under the

Method	Roughness	Albedo			Aligned Albedo			View Synthsis			Relighting		
	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRFactor [32]	-	19.4858	0.8641	0.2060	22.9647	0.9064	0.1617	22.7953	0.9168	0.1512	21.5373	0.8749	0.1708
PhySG* [29]	0.2682	21.2690	0.9722	0.0962	21.7968	0.9733	0.1845	23.4154	0.9871	0.0684	22.6288	0.9734	0.0726
Ours	**0.0723**	**24.1608**	**0.9782**	**0.0566**	**25.2511**	**0.9825**	**0.0581**	26.1918	0.9905	0.0438	**25.5934**	**0.9840**	**0.0410**
w/o vis. & ind. illum.	0.1575	23.3332	0.9758	0.0674	24.0401	0.9720	0.0679	**26.4971**	0.9923	0.0437	25.3919	0.9804	0.0451
w/o ind. illum.	0.0845	23.7422	0.9731	0.0677	24.6547	0.9819	0.0651	26.3454	**0.9927**	**0.0435**	25.4957	0.9836	0.0444
w/o latent space	0.0783	24.0930	0.9775	0.0593	25.2283	0.9824	0.0598	26.1846	0.9902	0.0449	25.5101	0.9837	0.0422

Table 1. **Quantitative evaluations.** We present the average results on the test images of all four synthetic scenes. "Aligned Albedo" refers to scaling the albedo prediction for each RGB channel to match the ground truth before computing the errors. We slightly modify PhySG [29] to adapt it to our data by outputting spatially varying roughness from an MLP instead of treating it as a global variable. Compared with previous methods and baseline models, our full model achieves the best performance in SVBRDF recovery and relighting. The view synthesis quality of the full model is slightly worse than the baselines, likely due to the rendering noise introduced by the visibility sampling.

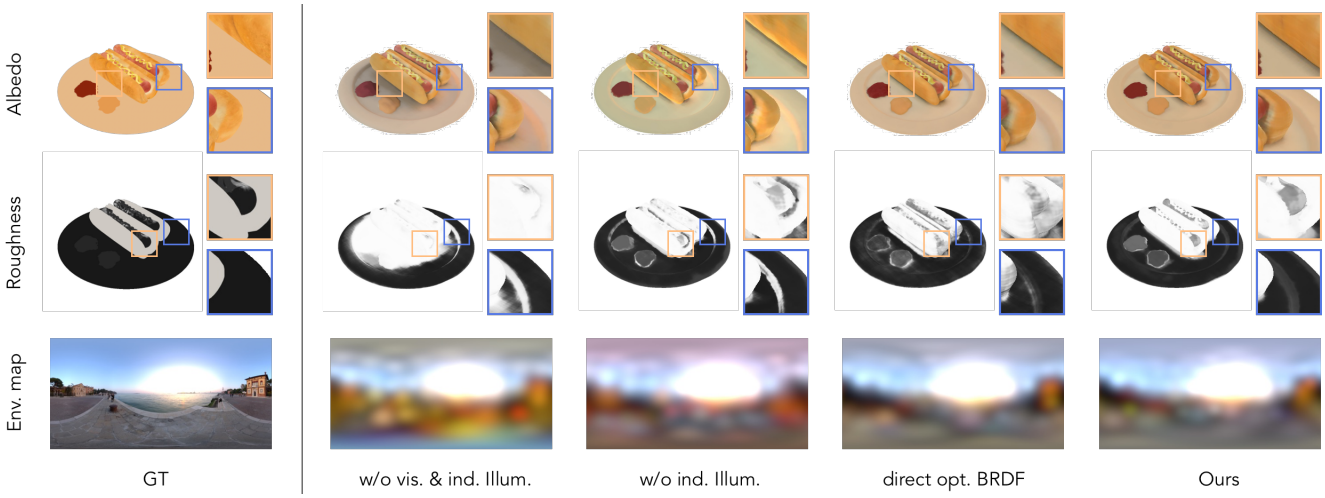


Figure 5. **Ablation study on a synthetic scene (hotdog).** Please refer to Section 4.3 for detailed descriptions.

assumption that all the surface points share the same environment lighting. It does not involve indirect illumination and visibility factors. It's not surprising that this variant performs worst for inverse rendering and relighting tasks. However, it yields a slight improvement in novel view synthesis. A possible reason is that visibility sampling introduces some rendering noise. "w/o ind. illum." produces unexpected brighter environment lighting and albedo compared to ground truth. That means, without modeling the indirect illumination, these indirect lighting effects would be baked into the estimated albedo by mistake. The "w/o latent space" variant trains an MLP that directly maps a 3D location to its diffuse and roughness using a re-rendering loss only, without latent space assumption (See Sec 3.4). The visualization shows that optimizing each surface point independently yields noise roughness.

4.4. Results on Real Captures.

We select 4 real objects made of various materials, such as plastic and leather, and capture them with a mobile phone moving around the upper hemisphere. The camera poses are estimated by COLMAP [20]. For each object, we uniformly sample 100 frames from the video and apply inverse gamma correction ($\gamma = 2.2$) to the images for training. Note that the environment may not be exactly ideal, as not all light is infinite distance, especially when capturing object-centric video indoors, and moving people will cast shadows on objects. Figure 6 shows the inverse rendering and relighting results. Our approach is able to infer plausible SVBRDF and support realistic relighting. See supplementary video for more results.

5. Conclusion

In this paper, we present a novel approach to efficiently modeling the indirect illumination in the inverse rendering task. Most of the previous methods have not considered indirect illumination since simulating it is intractable within a physically-based rendering framework. Instead, we utilize the neural outgoing radiance field and derive indirect illu-

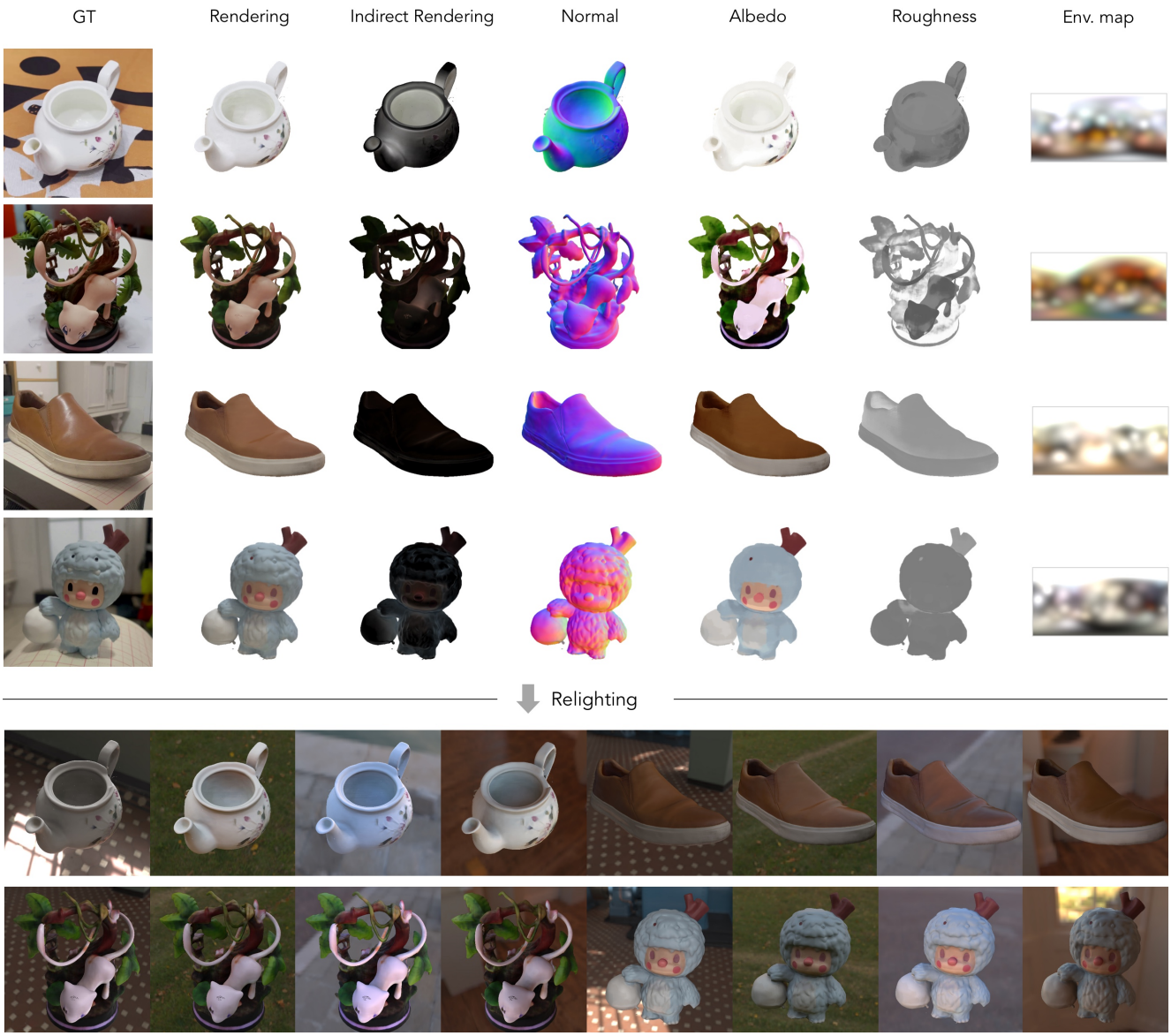


Figure 6. **Results on real captures.** Our method is capable of dealing with real-world objects composed of multiple materials. For each captured object, we show an image in the test set, our rendering, rendering under indirect illumination with our estimated shape and SVBRDF, decomposed normal, albedo and roughness. With decomposed factors, we can relight the object under arbitrary lighting. Here we show the results under four novel real-world illuminations.

mination from it. We demonstrate that, together with our proposed BRDF prior and SG-based visibility estimation, the full pipeline is able to estimate high-quality albedo and roughness from multi-view images captured under natural illumination and support realistic relighting.

Limitations. Our approach has the following limitations. First, our pipeline strongly relies on fine geometry as an input. We cannot deal with the case where the geometry fails to be reconstructed using [27]. Fortunately, our rendering model can be easily migrated to other surface-based geometric representations. Second, we parameterize BRDF with fixed $F_0 = 0.02$ in the Fresnel term [6]. In other words, we assume that the recovered materials are dielectric. Making F_0 learnable would exacerbate the ambiguity of the inverse problem. Learning-based prior or extra observations can help alleviate this ambiguity. We leave this as future work.

Acknowledgements: This work was supported by NSFC (No. 62172364) and Alibaba Group through Alibaba Innovative Research Program.

References

[1] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1670–1687, 2015. 2

[2] Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milovs Havsan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *ArXiv*, abs/2008.03824, 2020. 1, 2, 5

[3] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Milovs Havsan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *ECCV*, 2020. 1, 2

[4] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David J. Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5959–5968, 2020. 2

[5] Mark Boss, Raphael Braun, V. Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. *ArXiv*, abs/2012.03918, 2020. 1, 2

[6] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7. vol. 2012, 2012. 4, 8

[7] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion. *ACM Transactions on Graphics (TOG)*, 33:1 – 12, 2014. 1, 2

[8] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (TOG)*, 38(6):1–19, 2019. 1

[9] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 3

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[11] Hendrik PA Lensch, Jochen Lang, Asla M Sá, and Hans-Peter Seidel. Planned sampling of spatially varying brdfs. In *Computer graphics forum*, volume 22, pages 473–482. Wiley Online Library, 2003. 1

[12] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2481, 2020. 2

[13] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37:1 – 11, 2018. 2

[14] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W. Jacobs. Shape and material capture at home. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6119–6129, 2021. 2

[15] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 5

[16] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H. Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Transactions on Graphics (TOG)*, 37:1 – 12, 2018. 2

[17] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011. 4

[18] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *ECCV*, 2020. 2

[19] Carolin Schmitt, Simon Donné, Gernot Riegler, Vladlen Koltun, and Andreas Geiger. On joint estimation of pose, geometry and svbrdf from a handheld scanner. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3490–3500, 2020. 2

[20] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[21] 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. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8597–8606, 2019. 2

[22] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7491–7500, 2021. 1, 2

[23] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. 2009. 3

[24] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[25] Xin Wei, Guojun Chen, Yue Dong, Stephen Ching-Feng Lin, and Xin Tong. Object-based illumination estimation with rendering-aware neural networks. In *ECCV*, 2020. 2

[26] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics (TOG)*, 35:1 – 12, 2016. 1, 2

[27] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 4, 5, 8

[28] Ye Yu and W. Smith. Inverserendernet: Learning single image inverse rendering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3150–3159, 2019. 2

[29] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 5, 7

[30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[31] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 40(1):1–17, 2021. 1

[32] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *arXiv preprint arXiv:2106.01970*, 2021. 1, 2, 5, 6, 7