Neural-PBIR Reconstruction of Shape, Material, and Illumination

Cheng Sun[1,2*] Guangyan Cai[1,3*] Zhengqin Li[1] Kai Yan[3] Cheng Zhang[1]
Carl Marshall[1] Jia-Bin Huang[1,4] Shuang Zhao[3] Zhao Dong[1]
[1]Meta RLR [2]National Tsing Hua University [3]University of California, Irvine [4]University of Maryland, College Park
[*]The authors contribute equally to this paper

(a) Input images (b1) Albedo (b2) Roughness (b3) Shape & lighting (c1) Novel view (c2) Relit 1 (c3) Relit 2

Figure 1: **Neural-PBIR** recovers high-fidelity material (b1,2), shape and lighting (b3), enabling realistic re-rendering (c1-3).

Abstract

Reconstructing the shape and spatially varying surface appearances of a physical-world object as well as its surrounding illumination based on 2D images (e.g., photographs) of the object has been a long-standing problem in computer vision and graphics. In this paper, we introduce an accurate and highly efficient object reconstruction pipeline combining neural based object reconstruction *and* physics-based inverse rendering (PBIR). *Our pipeline firstly leverages a neural SDF based shape reconstruction to produce high-quality but potentially imperfect object shape. Then, we introduce a neural material and lighting* distillation *stage to achieve high-quality predictions for material and illumination. In the last stage, initialized by the neural predictions, we perform PBIR to refine the initial results and obtain the final high-quality reconstruction of object shape, material, and illumination. Experimental results demonstrate our pipeline significantly outperforms existing methods quality-wise and performance-wise. Code:* https://neural-pbir.github.io/

1. Introduction

Reconstructing geometry, material reflectance, and lighting from images, also known as inverse rendering, is a long-standing challenge in computer vision and graphics. Conventionally, the acquisition of the three intrinsic components has been mainly studied independently. For instance, multiview-stereo (MVS) [26, 27, 9] and time-of-flight [44] methods only focus on recovering object geometry, usually based on diffuse reflectance assumption. Classical material acquisition methods typically assume known or simple geometries (e.g., a planar surface) with highly controlled illuminations [22, 34, 43], usually created with a light stage or gantry. This significantly limits their practicality when such capturing conditions are unavailable.

Recently, the advent of novel techniques enables us to jointly reconstruct shape, material, and lighting from 2D images of an object. At a high level, these techniques can be classified into two categories. Neural reconstruction methods encode the appearance of objects into a multi-layer perceptron (MLP) and optimize the network by minimizing the rendering errors from different views through differentiable volume ray tracing. NeRF [18] reconstructs a density field-based radiance field that allows high-quality view synthesis but not relighting. A series of methods [31, 35, 24] compute the density field from the signed distance function to achieve high-quality geometry reconstruction. Recent works [40, 41, 3, 4, 20, 11, 42] seek to fully decompose shapes, materials, and lighting from input images. However, due to the high computational cost of volume ray tracing and neural rendering, those methods take hours [20, 11] or a day [3, 42] to run and usually cannot model more complex indirect illumination [40, 41, 3, 4, 20, 11], causing shadows and color bleeding to be baked into the material reflectance. Several new methods [28, 19, 7] significantly reduce the computational cost of radiance field reconstruction using hybrid volume representations and efficient MLPs. We are among the first that adopt these novel techniques for efficient joint recovery of geometry, material, and lighting.

On the contrary, *physics-based inverse rendering* (PBIR)

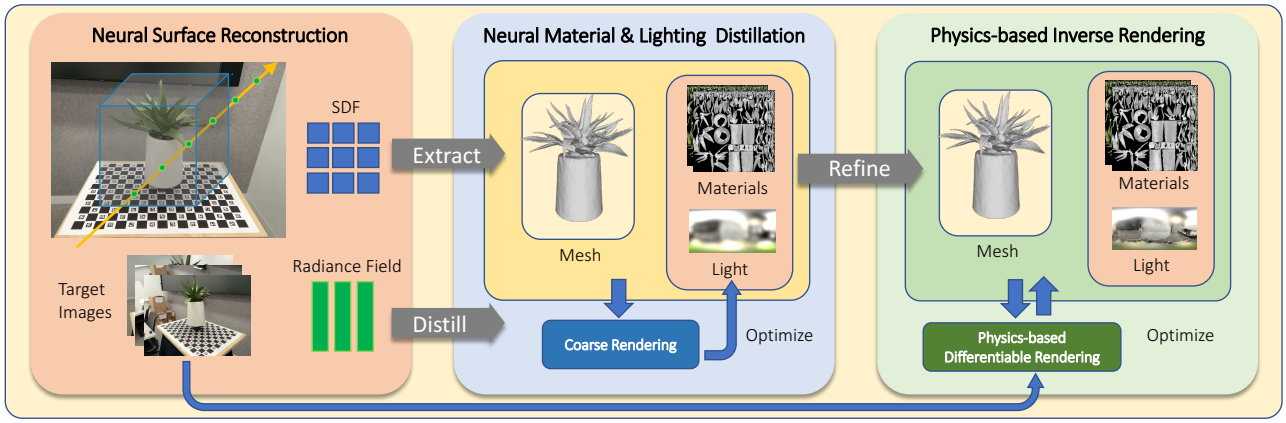


Figure 2: **Our pipeline for joint shape, material, and lighting estimation.**

[6, 16, 23, 1] optimizes shape, material, and lighting by computing unbiased gradients of image appearance with respect to scene parameters. Leveraging physics-based differentiable renderers [38, 12, 15], state-of-the-art PBIR pipelines can efficiently handle complex light transport effects such as soft shadow and interreflection. Such complex light transport effects cannot be easily handled through volume-based neural rendering. On the other hand, since PBIR methods rely on gradient-based optimization to refine intrinsic components, they can be prone to local minima and overfitting. Therefore, they may require a good initialization to achieve optimal reconstruction quality.

In this paper, we present a highly efficient and accurate inverse rendering pipeline with the advantages of both neural reconstruction and PBIR methods. Our pipeline attempt to estimate geometry, spatially varying material reflectance, and an HDR environment map from multiple images of an object captured under static but arbitrary lighting. As shown in Fig. 2, our pipeline consists of three stages. In the first stage, we propose a hybrid neural volume-based method for fast neural SDF and radiance field reconstruction, which achieves state-of-the-art geometry accuracy and efficiency. In the next stage, based on the reconstructed geometry and radiance field, we design an efficient optimization method to distill materials and lighting by fitting the surface light field. Our method relies on a radiance field to handle visibility and indirect illumination but avoids expensive volume ray tracing to significant computational cost compared to some recent works. Finally, we use an advanced PBIR framework [38, 12] to jointly refine the geometry, materials, and lighting. Note that our PBIR framework models complex light transport effects such as visibility, occlusion, soft shadows and interreflection in a physically correct and unbiased way while still being much faster than recent inverse rendering methods.

Concretely, our contributions include the following:

- A hybrid volume representation for fast and accurate geometry reconstruction.
- A efficient optimization scheme to distill high-quality initial material and lighting estimation from the reconstructed geometry and radiance field.
- An advanced PBIR framework that jointly optimizes materials, lighting and geometry with visibility and interreflection handled in a physically unbiased way.
- A end-to-end pipeline that achieves state-of-the-art geometry, material and lighting estimation that enables realistic view synthesis and relighting.

2. Related Works

Volumetric surface reconstruction. Recent progress in volumetric-based surface reconstruction of a static scene shows high-quality and robust results. NeuS [31] and VolSDF [35] replace NeRF [18]'s density prediction with signed distance values and proposes an unbiased and occlusion-aware volume rendering, achieving promising results. Subsequent works improve quality by introducing regularization losses to encourage surface smoothness [39], multiview consistency [8], and manhattan alignment [10]. These methods use a large MLP as their volumetric representation, which is however slow (many hours) to optimize per scene. Recent advances [28, 36, 19, 7] show great optimization acceleration without loss of quality by using an explicit grid. Unfortunately, using explicit volume to model SDF leads to bumpy and noisy surfaces [37]. Supervisions from SfM sparse point cloud [39, 8] or monocular depth/normal [37] may mitigate the difficulty but it's out of our scope. We propose simple and effective regularization losses for explicit SDF grid optimization, achieving fast and high-quality surfaces without using external priors.

Material and lighting estimation. Several neural reconstruction methods [40, 41, 3, 4, 20, 11, 42] adopt the same setting as ours to simultaneously reconstruct geometry, material, and lighting from multiple images. An earlier work [41] directly optimizes a low-resolution environment map and materials from fixed geometry, leading to noisy light-

ing reconstruction caused by the highly ill-posed nature of this problem. More recent methods model lighting with a mixture of spherical Gaussians [40, 3, 42] or pre-filtered approximation [4, 20], and constrain material reflection with a spherical Gaussian lobe [40] or a low dimensional latent code [3, 4, 42] to improve the reconstruction quality. Most of them only consider direct illumination without occlusion [40, 3, 20] or use a fixed shadow map [41, 4] due to the high computational cost of MLP-based neural rendering. Nvdiffrecmc [11] models shadows using a differentiable ray tracer with a denoiser that significantly reduces the number of samples but still cannot model interreflection. MII [42] trains another network to predict visibility and interreflection from the radiance field, which takes several hours. On the contrary, our hybrid, highly optimized neural SDF framework enables us to efficiently extract geometry, materials, lighting, visibility, and interreflection in less than 10 minutes. Our advanced PBIR framework for the first time allows holistic optimization of lighting, material, and geometry with direct and indirect illumination modeled in a physically unbiased way.

Physics-based inverse rendering. Different from neural-based methods, several recent works [16, 11, 6] utilize classic image formulation models developed in the graphics community and try to inverse this process using gradient-based optimizations, with the gradients computed using physics-based differentiable renderers. These approaches can handle complex light transport effects during optimization but are prone to local minima and overfitting. In our work, instead of optimizing from scratch, we perform PBIR as a refinement stage for better robustness.

3. Our Method

Provided multi-view images of an opaque object under fixed unknown illumination (with known camera parameters), our technique reconstructs the shape and reflectance of the object as well as the illumination condition.

As illustrated in Fig. 2, our pipeline is comprised of three main stages. The first stage (Sec. 3.1) is *a fast and precise surface reconstruction step* that brings direct SDF grid optimization into NeuS [31]. Associated with this surface is an overfitted radiance field that does not fully model the surface reflectance of the object. Our second stage (Sec. 3.2) is *an efficient neural distillation method* that converts the radiance fields to physics-based reflectance [5] and illumination models. Lastly, our third stage (Sec. 3.3) utilizes *physics-based inverse rendering* (PBIR) to further refine the object geometry and reflectance reconstructed by the first two stages. This stage leverages physics-based differentiable rendering that captures global illumination (GI) effects such as soft shadows and interreflection.

3.1. Neural Surface Reconstruction

Taking as input multiple images of an opaque object under fixed illumination (and with known camera parameters), the first stage of our method produces a detailed reconstruction of the object's surface. To this end, we optimize the object surface and an outgoing radiance field that best describes the input images. In this stage, for efficiency and robustness, we express the object surface as the zero-level set $\{\boldsymbol{x} \in \mathbb{R}^3 \mid S(\boldsymbol{x}) = 0\}$ of a signed distance field (SDF) $S(\boldsymbol{x})$, and the radiance field $L_\mathrm{o}(\boldsymbol{x}, -\boldsymbol{v})$ (for any position $\boldsymbol{x} \in \mathbb{R}^3$ and viewing direction $\boldsymbol{v} \in \mathbb{S}^2$) as a non-physics-based general function approximator.

Unbiased volume rendering with SDF. To compute the color of a pixel, we first sample N query points $\{\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{v}\}_{i=1}^N$ (with $0 < t_1 < t_2 < \ldots < t_N$) along the corresponding camera ray originated at the camera's location $\boldsymbol{o}$ with viewing direction $\boldsymbol{v}$. We then query the scene representation for points sign distance $S(\boldsymbol{x}_i)$ and radiance $L_\mathrm{o}(\boldsymbol{x}_i, \boldsymbol{v})$. Following NeuS [31]'s unbiased rendering, we activate the queried signed distance into alpha for all $1 \le i \le N$ via

$$\alpha_i = \max\left(0,\ \frac{\sigma(S(\boldsymbol{x}_i)) - \sigma(S(\boldsymbol{x}_{i+1}))}{\sigma(S(\boldsymbol{x}_i))}\right), \tag{1}$$

where σ denotes the Sigmoid function. Then, the pixel color C is computed by the alpha blending of the queried alphas and radiance values

$$C = \sum_{i=1}^{N} T_i\,\alpha_i\,L_\mathrm{o}(\boldsymbol{x}_i, -\boldsymbol{v}) \quad \text{where } T_i = \prod_{j=1}^{i-1}(1-\alpha_j). \tag{2}$$

Voxel-based scene representation. NeuS uses a large MLP to model the SDF S and radiance field L_o. Unfortunately, since this MLP is expensive to query, NeuS' optimization processes can be very time-consuming.

For better performance, we adapt the dense-grid-based scene representation from DVGO [28] by setting

$$\begin{aligned} S(\boldsymbol{x}) &= \mathrm{interp}(\boldsymbol{x},\ V^{(\mathrm{sdf})})\,, \\ L_\mathrm{o}(\boldsymbol{x}, -\boldsymbol{v}) &= \mathrm{MLP}\left(\mathrm{interp}(\boldsymbol{x},\ V^{(\mathrm{feat})}),\ \boldsymbol{v}\right), \end{aligned} \tag{3}$$

where $\mathrm{interp}()$ indicates trilinear interpolation, $V^{(\mathrm{sdf})}$ is a dense SDF grid, $V^{(\mathrm{feat})}$ is a dense feature grid, and MLP is a single-hidden-layer MLP with ReLU activation.

In practice, we use two sets of $V^{(\mathrm{sdf})}$ and $V^{(\mathrm{feat})}$ grids to model the foreground (i.e., the object of interest) and the background with varying resolutions. Please refer to the supplement for more details.

Adaptive Huber loss. We incorporate a Huber loss to reduce the impact of specular highlights which can cause bumpy artifacts in reconstructed surfaces:

$$\mathcal{L}_\mathrm{photo} = \sum_r \begin{cases} (\mathcal{I}[r] - C[r])^2, & (|\mathcal{I}[r] - C[r]| \le t) \\ 2t\,|\mathcal{I}[r] - C[r]| - t^2, & (\text{otherwise}) \end{cases} \tag{4}$$

where: $\mathcal{I}[r]$ and $C[r]$ denote, respectively, the colors of the r-th pixels of the target and rendered images; $t \in \mathbb{R}_{>0}$ is a hyper-parameter.

Intuitively, the infinity norm of the gradient of $\mathcal{L}_{\text{photo}}$ is clamped by $2t$ to prevent bright pixels (*e.g.*, those exhibiting specular highlights) dominating the optimization. In practice, instead of tuning a constant t, we adaptively update t by the running mean of the median of $|\mathcal{I}[r] - C[r]|$ in each iteration to clamp gradient for about half of the pixels.

Laplacian regularization. To further improve the robustness of our approach and reduce potential artifacts, we regularize our SDF grid $V^{(\text{sdf})}$ using a Laplacian loss:

$$\mathcal{L}_{\text{lap}} = \sum_{u} \left(\sum_{u' \in N(u)} \left(V^{(\text{sdf})}[u'] - V^{(\text{sdf})}[u] \right) \right)^2 , \quad (5)$$

where $V^{(\text{sdf})}[u]$ indicates the SDF value stored at the grid point u, and $N(u)$ denotes the six direct neighbors of u.

Training. Following DVGO [28], we use progressive grid scaling for efficiency and more coherent results. Our experiments also indicate that using a per-point RGB loss $\mathcal{L}_{\text{pp\_rgb}}$ improves convergence speed and quality:

$$\mathcal{L}_{\text{pp\_rgb}} = \sum_{r,i} T_i^{(r)} \alpha_i^{(r)} \left| L_{\text{o}} \left(\boldsymbol{x}_i^{(r)}, -\boldsymbol{v}^{(r)} \right) - \mathcal{I}[r] \right| , \quad (6)$$

where $\alpha_i^{(r)}$ and $T_i^{(r)}$ are computed using Eqs. (1) and (2) for each pixel r. In summary, to train our SDF and feature grids $V^{(\text{sdf})}$ and $V^{(\text{feat})}$ in Eq. (3), we minimize the objective:

$$\mathcal{L}_{\text{surf}} = \mathcal{L}_{\text{photo}} + w_{\text{lap}} \mathcal{L}_{\text{lap}} + w_{\text{pp\_rgb}} \mathcal{L}_{\text{pp\_rgb}} . \quad (7)$$

Postprocessing. Optionally, our surface reconstruction stage generates for each input image an anti-aliased mask that separates the object from the background using mesh rasterisation and alpha matting [25]. The generated masks can be used by our physics-based inverse rendering stage (Sec. 3.3) to facilitate the refinement of object shapes.

3.2. Neural material and lighting distillation

Provided the optimized object surface and outgoing radiance field from the surface reconstruction stage (Sec. 3.1), the goal of the second stage of our pipeline is to obtain an initial estimation of surface reflectance and illumination condition. To this end, we leverage the radiance field L_{o} as the teacher model to distill the learned surface color into physics-based material and illumination models (that can be rendered to reproduce L_{o}) as follows.

As preprocessing for this stage, we extract a triangle mesh $M^{(0)}$ from the optimized SDF $S(\boldsymbol{x})$ given by Eq. (3) using (non-differentiable) marching cube. Additionally, for each mesh vertex v, we compute its normal $M_{\boldsymbol{n}}[v] \in \mathbb{S}^2$ based on the gradient ∇S of the SDF.

Material and illumination models. To model spatially varying surface reflectance, we use the widely-adopted Disney microfacet BRDF [14] parameterized by surface albedo and roughness. In practice, instead of using a large MLP to model the spatially varying BRDF parameters [40, 41, 20, 42], we store the albedo $M_{\text{a}}[v] \in [0, 1)$ and roughness $M_{\text{r}}[v] \in \mathbb{R}_{>0}$ for each vertex v of the extracted mesh $M^{(0)}$ (and interpolate them in the interior of triangle faces) for better performance. To model environmental illumination, we use mixtures of spherical Gaussians [30, 17, 40, 42] with 256 lobes.

Coarse rendering. For fast training, we opt to use numerical integration on stratified pre-sampled light directions Ω for efficiency (we use 256 samples in practice). Specifically, for each vertex v of the extracted mesh $M^{(0)}$ and light direction $\boldsymbol{\omega}_{\text{i}} \in \Omega$, we precompute the visibility $\text{Vis}[v, \boldsymbol{\omega}_{\text{i}}]$ and indirect illumination $L_{\text{i}}^{(\text{ind})}[v, \boldsymbol{\omega}_{\text{i}}]$ by tracing a ray from the vertex in the direction $\boldsymbol{\omega}_{\text{i}}$. If this ray intersects the object surface at some $\boldsymbol{x}$, we set $\text{Vis}[v, \boldsymbol{\omega}_{\text{i}}] = 0$ and $L_{\text{i}}^{(\text{ind})}[v, \boldsymbol{\omega}_{\text{i}}] = L_{\text{o}}(\boldsymbol{x}, -\boldsymbol{\omega}_{\text{i}})$ [42]. Otherwise, we set $\text{Vis}[v, \boldsymbol{\omega}_{\text{i}}] = 1$. Then, the incident radiance at each vertex can be expressed as

$$L_{\text{i}}(\boldsymbol{\omega}_{\text{i}}) = L_{\text{env}}^{\text{SG}}(\boldsymbol{\omega}_{\text{i}}) \, \text{Vis}[\boldsymbol{\omega}_{\text{i}}] + L_{\text{i}}^{(\text{ind})}[\boldsymbol{\omega}_{\text{i}}] \, (1 - \text{Vis}[\boldsymbol{\omega}_{\text{i}}]) , \quad (8)$$

where $L_{\text{env}}^{\text{SG}}$ is the SG-based environmental illumination. Lastly, the outgoing radiance is computed by

$$\hat{L}_{\text{o}}(\boldsymbol{\omega}_{\text{o}}) = \frac{1}{Z} \sum_{\boldsymbol{\omega}_{\text{i}} \in \Omega} L_{\text{i}}(\boldsymbol{\omega}_{\text{i}}) \, f(\boldsymbol{\omega}_{\text{i}}, \, \boldsymbol{\omega}_{\text{o}}, \, M_{\text{n}}) \, (M_{\text{n}} \cdot \boldsymbol{\omega}_{\text{i}}) , \quad (9)$$

where Z is a normalization factor, f denotes the BRDF function parameterized by M_{a} and M_{r}. In Eqs. (8) and (9), we omit dependencies on vertex v for brevity.

Training. In each iteration, we randomly sample a outgoing direction $\boldsymbol{\omega}_{\text{o}}$ (with $M_{\text{n}}[v] \cdot \boldsymbol{\omega}_{\text{o}}[v] > 0$) for each vertex v and minimize the absolute difference with the teacher model:

$$\mathcal{L}_{\text{distill}} = \sum_{v} \left| \hat{L}_{\text{o}} \left(v, \boldsymbol{\omega}_{\text{o}}[v] \right) - L_{\text{o}} \left(v, \boldsymbol{\omega}_{\text{o}}[v] \right) \right| . \quad (10)$$

To encourage smoothness, we apply total variation loss on the per-vertex albedo M_{a} and roughness M_{r}:

$$\mathcal{L}_{\text{v\_reg}} = \sum_{(v_1, v_2) \in \text{edge}} \sum_{* \in \{\text{a,r}\}} |M_*[v_1] - M_*[v_2]| . \quad (11)$$

Also, we regularize the SG-based illumination $L_{\text{env}}^{\text{SG}}$ using

$$\mathcal{L}_{\text{bg}} = \sum_{\boldsymbol{\omega}_{\text{i}} \in \Omega} \left| L_{\text{env}}^{\text{SG}}(\boldsymbol{\omega}_{\text{i}}) - (L_{\text{env}}^{\text{SG}})'(\boldsymbol{\omega}_{\text{i}}) \right| , \quad (12)$$

where $(L_{\text{env}}^{\text{SG}})'$ indicates the averaged background observation (see the supplement for more details).

In summary, we optimize per-vertex albedo M_{a}, roughness M_{r}, and the environmental illumination $L_{\text{env}}^{\text{SG}}$ by minimizing

$$\mathcal{L}_{\text{distill\_total}} = \mathcal{L}_{\text{distill}} + w_{\text{v\_reg}} \mathcal{L}_{\text{v\_reg}} + w_{\text{bg}} \mathcal{L}_{\text{bg}} . \quad (13)$$

Postprocessing. With the per-vertex attributes M_{a} and M_{r} obtained, we UV-parameterize the extracted mesh $M^{(0)}$ and pixelize M_{a} and M_{r}, respectively, into an ableo map $T_{\text{a}}^{(0)}$ and a roughness map $T_{\text{r}}^{(0)}$. These maps will be refined by our following inverse-rendering stage (Sec. 3.3).

3.3. Physics-Based Inverse Rendering

Reconstructions produced by our shape reconstruction and neural material distillation stages (Secs. 3.1 and 3.2) comprise three components: (i) environmental illumination $L_{\text{env}}^{(0)} := L_{\text{env}}^{\text{SG}}$ expressed as spherical Gaussians; (ii) object reflectance parameterized with an albedo map $T_{\text{a}}^{(0)}$ and a roughness map $T_{\text{r}}^{(0)}$; and (iii) object surface mesh $M^{(0)}$.

Despite being high-fidelity, these reconstructions can still lack sharp details and are not immune to artifacts (see Fig. 7). To further improve reconstruction quality, we utilize a physics-based inverse rendering stage. Specifically, initialized our neural reconstructions (i.e., $L_{\text{env}}^{(0)}$, $T_{\text{a}}^{(0)}$, $T_{\text{r}}^{(0)}$, and $M^{(0)}$), we apply gradient-based optimization to minimize the *inverse-rendering loss*:

$$\mathcal{L}_{\text{IR}} = \mathcal{L}_{\text{img}} + w_{\text{mask}}\,\mathcal{L}_{\text{mask}} + w_{\text{reg}}\,\mathcal{L}_{\text{reg}}, \tag{14}$$

where

$$\mathcal{L}_{\text{img}}(L_{\text{env}}, T_{\text{a}}, T_{\text{r}}, M) = \sum_j \left\| \mathcal{I}_j - \mathcal{R}_j(L_{\text{env}}, T_{\text{a}}, T_{\text{r}}, M) \right\|_1, \tag{15}$$

is the *image loss* given by the sum of L1 losses between the j-th target image $\mathcal{I}_j$ and the corresponding rendered image $\mathcal{R}_j(L_{\text{env}}, T_{\text{a}}, T_{\text{r}}, M)$. Additionally, $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{mask}}$ in Eq. (14) denote, respectively, the *regularization* and the *mask* losses—which we will discuss in the following.

Environment map optimization. We recall that the initial illumination $L_{\text{env}}^{(0)} = L_{\text{env}}^{\text{SG}}$ is a coarse reconstruction expressed as a set of spherical Gaussians (SGs). To further refine it, we employ a two-step process as follows. In the first step, we directly optimize the SG parameters (i.e., per-lobe means and variances). In the second step, we pixelize the optimized SG representation into an environment map (using the latitude-longitude parameterization) and perform per-pixel optimization.

SVBRDF optimization. Since our initializations $T_{\text{a}}^{(0)}$ and $T_{\text{r}}^{(0)}$ are already high-quality, we perform per-texel optimization for the albedo and roughness maps T_{a} and T_{r}. Further, to make our optimization less prune to Monte Carlo noises produced by our physics-based renderer (discussed later), we regularize T_{r} using a total variation loss:

$$\mathcal{L}_{\text{reg}}(T_{\text{r}}) = \sum_{(x,y)} \sum_{(x',y')} \left| T_{\text{r}}[x', y'] - T_{\text{r}}[x, y] \right|, \tag{16}$$

where $T_{\text{r}}[x, y]$ denotes the (x, y)-th texel of the roughness map T_{r}, and $(x', y') \in \{(x+1, y),\ (x, y+1)\}$ are two direct neighbors of (x, y).

Shape refinement. In all our experiments, object geometries predicted by our neural stages accurately recover object topology. Thus, although it is possible to directly optimize SDFs in inverse rendering [2, 29], we opt to use explicit mesh-based representations for the object surface M in this stage for better efficiency. To make our per-vertex optimization more robust to Monte Carlo noises, we utilize Nicolet et al.'s AdamUniform optimizer [21].

To further improve efficiency, we leverage object masks produced either as input or by our surface reconstruction (Sec. 3.1) and introduce a mask loss:

$$\mathcal{L}_{\text{mask}}(M) = \sum_j \left\| S_j - \mathcal{R}_j^{\text{mask}}(M) \right\|_1, \tag{17}$$

where S_j is our predicted mask for the j-th target image and $\mathcal{R}_j^{\text{mask}}$ the corresponding rendered mask.

Differentiable rendering. To differentiate the image and the mask losses defined in Eqs. (15) and (17), we develop a physics-based Monte Carlo differentiable renderer that implements path-space differentiable rendering [38] and utilizes state-of-the-art numerical backend [12].

Most, if not all, previous techniques including MII [42] and nvdiffrec-mc [11] rely on highly simplified differentiable rendering processes that typically neglect global-illumination (GI) effects and produce biased gradients with respect to geometry. On the contrary, our differentiable renderer offers unbiased gradients and the generality of differentiating GI (and anti-aliased masks) with respect to surface geometries. Consequently, our pipeline is capable of generating higher-quality reconstructions than state-of-the-art methods, which we will demonstrate in Sec. 4.

4. Results

To demonstrate the effectiveness of our method, we show our reconstructions on synthetic input images and captured photographs in Secs. 4.1 and 4.2, respectively. We compare reconstructions obtained with our technique with two state-of-the-art baselines: MII [42] and nvdiffrec-mc [11]. Additionally, we conduct ablation studies to evaluate several components of our pipeline in Sec. 4.3. Please refer to the supplement for more results.

4.1. Synthetic Data

We assess the effectiveness of our proposed method using the synthetic dataset made available by MII [42]. This dataset comprises four virtual scenes—each of which includes multi-view renderings (with object masks) of an object under some natural environmental illumination (with the ground truth environment map provided). For each scene, the dataset provides 100 target images accompanied by camera poses and object masks for training. Additionally, the testing set consists of renderings of ground truth

Method	Speed	Relighting			Aligned albedo			Albedo			Rough.
	Time↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓
nvdiffrec-mc [11]	∼2 h	23.93	0.946	0.074	**29.72**	**0.959**	**0.057**	18.25	0.899	0.103	0.009
MII [42]	∼10 h	27.53	0.947	0.087	25.77	0.935	0.066	24.62	0.931	**0.064**	**0.008**
Ours - Distilled only	<15 m	30.26	0.961	0.059	27.67	0.933	0.079	26.20	0.931	0.093	0.009
Ours - Const. init.	∼ 37 m	30.25	**0.970**	0.050	28.55	0.940	0.070	25.83	0.940	0.080	0.010
Ours - w/o GI	∼45 m	30.57	0.960	0.050	27.71	0.940	0.070	26.38	0.940	0.082	0.009
Ours - w/o shape ref.	∼45 m	30.61	0.965	0.049	28.74	0.944	0.067	26.84	0.941	0.082	**0.008**
Ours - Full	∼1 h	**30.73**	0.966	**0.047**	29.06	0.946	0.067	**26.85**	**0.944**	0.080	**0.008**

Table 1: **Relighting, material reconstruction, and view-interpolation quality on MII dataset [42].** We compare our method with MII and Nvdiffrec-mc. The highest performing number is presented in bold, while the second best is underscored. For "Ours - Distilled only", we did not run the PBIR stage. For "Ours - Const. init." and "Ours - w/o GI", we did not run the shape refinement. Please refer to Secs. 4.1 and 4.3 for more details.

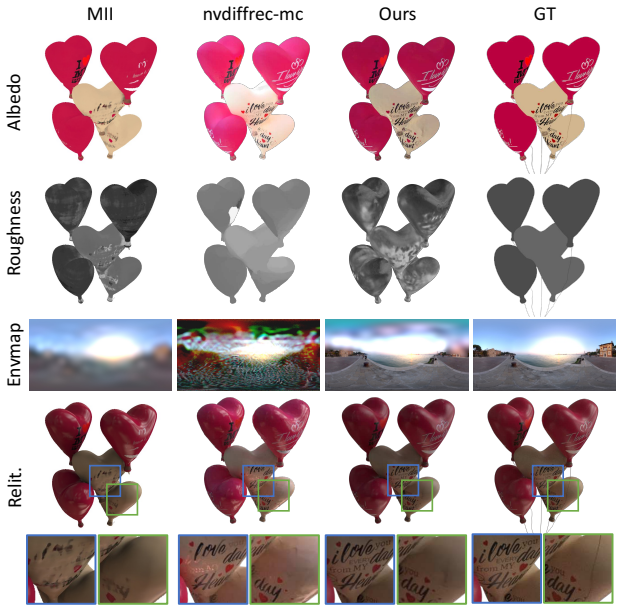


Figure 3: **Qualitative comparisons on the MII data.**

albedo, roughness, and the object under two novel lighting conditions in 200 poses. We apply our technique and the baselines to the posed training target images and masks (but not the GT environment maps).

Tab. 1 presents the quantitative comparisons where we compare our method's reconstructions with the baselines by rendering them in the testing poses and comparing them with the ground truth images. In line with MII, we report PSNR, SSIM, and LPIPS for the relit and albedo images, and the MSE for the roughness images averaged across all scenes. MII also reports error metrics on aligned albedo to reduce the impact of albedo-light ambiguity [42]. Specifically, we compute an RGB scale that minimizes the difference between the reconstructed albedo images and the ground truth albedo images, and reconstructed albedo images are scaled before evaluation. Our method outperforms the baselines and takes less time.

Methods	Novel-view		Captured light re-rendering			
			raw		aligned	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Nvdiffrecmc	30.3	0.94	21.8	0.92	28.0	0.94
MII	28.9	0.94	27.5	0.94	28.6	0.94
Ours	**31.6**	**0.96**	**28.8**	**0.95**	**30.7**	**0.95**

Table 2: **Quantitative comparison on Our Real Dataset.** To inspect material quality for relighting, we capture 360 images for our dataset and evaluate the rendering results under the captured lighting.

Figure 5: **Rerendering of reconstruction results under captured (GT) illumination.** We rescale all renderings to match the overall brightness of the GT image.

We show qualitative results in Fig. 3. Since MII imposes a sparsity constraint to reconstruct a sparse set of materials, the results tend to be over-blurred. Despite being able to produce sharp texture maps, nvdiffrec-mc suffers from inaccurate illumination reconstructions, causing their reconstructed albedo maps to be color-shifted. Compared with these two baselines, our method produces significantly more accurate material and illumination reconstructions.

4.2. Real Data

To further demonstrate the robustness of our method, we captured five real scenes under indoor lighting and again compared the reconstruction quality with MII and nvdiffrec-mc. For each scene, we use around 200 measured images for training and 10–20 for testing. We place a checkerboard beneath the object for the geometric calibration of camera

Figure 4: **Novel-view interpolation on our real dataset.** Our technique produces high-fidelity reconstructions with minimal artifacts. We report the average PSNR↑ and SSIM↑ below each image.

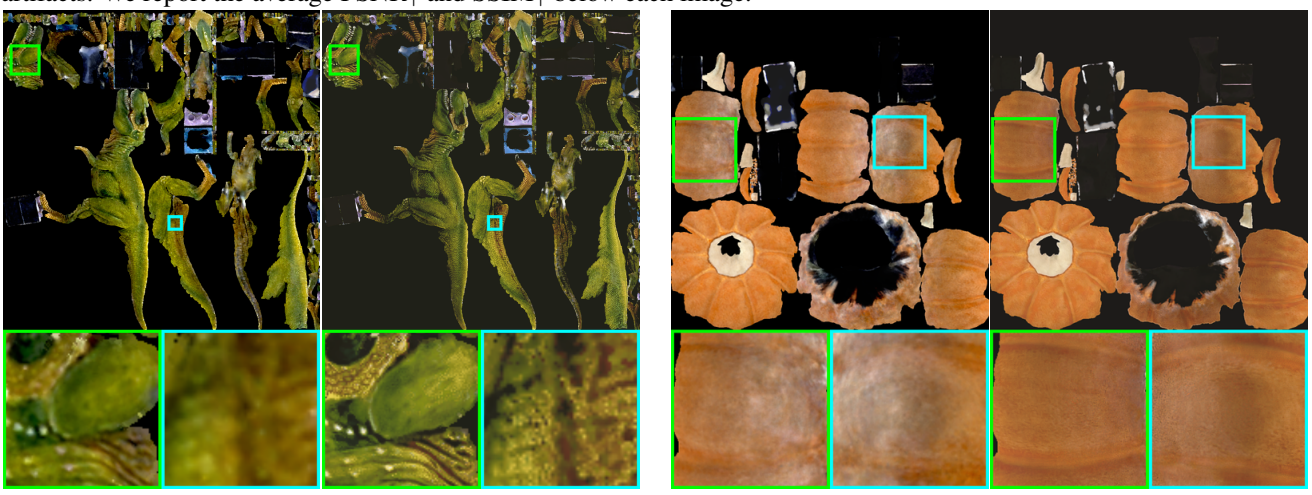


Figure 7: **Reconstructed albedo before and after PBIR.** Despite our neural distilling outperform previous arts by a large margin, we still observe blurriness and light baking in the reconstructed materials. Our PBIR stage can provide sharper details and remove light baking efficiently.

parameters. For each object, we manually annotate an object 3D bounding box to crop the reconstructed meshes to focus the comparisons on the object of interest. In addition, we measure ground truth environment map (which we use for evaluation only) for each scene using an HDR 360 camera. As there are no ground truth foreground masks while our baselines require them, we use masks generated in our surface reconstruction stage (§3.1).

We compare the qualities of reconstructions produced by our method and the baselines using novel-view renderings (since we do not have the GTs). As shown in Fig. 4, our reconstructions produce detailed renderings that closely re-

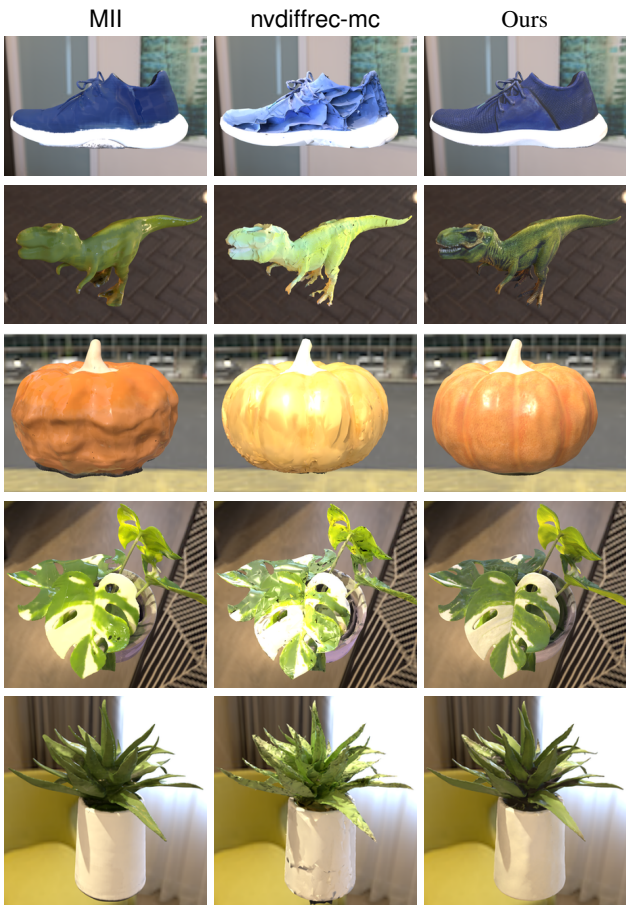


Figure 6: **Novel-illumination renderings of the reconstructed models.** Our results offer the highest overall quality with minimal artifacts.

semble the GTs, while MII and nvdiffrec-mc's results suffer from various artifacts (*e.g.* lacking details and bumpy surfaces). In addition, to evaluate the quality of reconstructed object shape and reflectance, we compare renderings under captured GT illumination in Fig. 5. The quantitative comparison is summarized in Tab. 2 where we achieve the best novel-view synthesis results under both the reconstructed and the captured environment maps. We show qualitative comparison under novel-illumination in Fig. 6.

4.3. Evaluations and Ablations

Surface reconstruction. To evaluate the quality of reconstructed geometries produced by our surface reconstruction stage (Sec. 3.1), we use the subsampled 15 scenes of DTU dataset [13]. Each scene has 49 or 64 images with camera parameters and masks provided. The quantitative results are concluded in Tab. 3. Our direct SDF grid optimization uses significantly less time than NeuS [31] while still outperforming NeuS surface accuracy measured in chamfer distance (CD). Our simple surface reconstruction with dense grid also achieves similar performance comparing to the most recent Voxurf [33] and NeuS2 [32].

Methods	COLMAP	NeuS	Voxurf	NeuS2	Our
Runtime↓	1 hrs	5.5 hrs	16 mins	**5 mins**	**5 mins**
CD (mm)↓	1.36	0.77	0.72	0.70	**0.66**

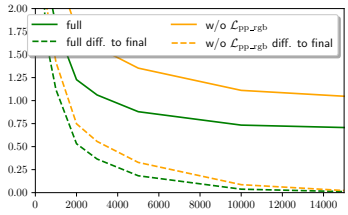
Table 3: **Surface reconstruction quality on DTU dataset [13].** The results are averaged across the 15 scenes. We include two recent works, Voxurf [33] and NeuS2 [32], for reference. See supp. for results breakdown.

$\mathcal{L}_{\text{lap}}$				✓		✓	✓	✓
$\mathcal{L}_{\text{pp\_rgb}}$			✓		✓		✓	✓
ada. huber		✓			✓	✓		✓
CD (mm)↓	1.50	1.37	1.24	1.11	1.03	1.00	0.89	**0.68**

Table 4: **Ablation study of SDF grid regularizations.** The results are averaged over the 15 objects on DTU dataset.

We conduct a comprehensive ablation study for the three losses that help us achieve direct SDF grid optimization. As shown in Tab. 4, naively adapting DVGO without regularization leads poor quality. Among the regularization, $\mathcal{L}_{\text{lap}}$ is the most significant, which enforces the SDF grid to evolve smoothly during optimization. Adding $\mathcal{L}_{\text{pp\_rgb}}$ and using adaptive Huber loss also shows good improvement. Combing all three losses offers the best results.

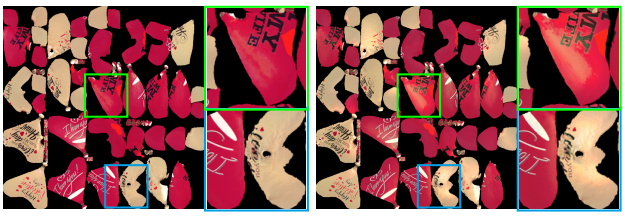
We also show the chamfer distance (CD) in the first 15k iterations with and without $\mathcal{L}_{\text{pp\_rgb}}$ here. The solid lines are CD, while the dashed lines are the difference to the final CD. The results are averaged over the 15 DTU scenes. Using $\mathcal{L}_{\text{pp\_rgb}}$ improves final quality and speeds up convergence to the final CD in fewer iterations.



Please note that our results on DTU dataset are directly from the shape reconstruction stage. We skip evaluating the shape refinement of our physics-based inverse rendering as DTU exhibit vary light occlusion from robot arms. Please see the supplement for more results.

Neural material distilling. We now demonstrate the usefulness of our neural material distilling stage (Sec. 3.2)—which provides high-quality material predictions used to initialize our PBIR stage (Sec. 3.3). Specifically, when using constant initializations for surface reflectance (with $T_{\text{a}} = 0.5$, $T_{\text{r}} = 0.5$) and illumination (with a gray environment map), it takes our PBIR pipeline 37 minutes (see the row labeled as "Ours - const. init." in Tab. 1) to generate reconstructions with a similar level of accuracy as the neural distilling stage does in less than 15 minutes (see the row labeled as "Ours - distilled only").

Importance of PBIR. The distilling method is efficient by sidestepping ray tracing and approximating global illu-



w/ GI (27.69 / 0.940) w/o GI (26.58 / 0.934)

Figure 8: **Reconstructed albedo with/without GI.** We show the impact of modeling global illumination (GI) on material reconstruction. The above images are the reconstructed albedo maps with GI on/off, with their errors listed below (PSNR↑ / SSIM↑). Without GI, the optimization tends to "bake" the indirect lighting into the albedo map.

mination but it sacrifices accuracy and tends to "bake" interreflection into the materials. We use our physically-based differentiable renderer to further refine the material. In 7, we show the albedo maps before and after PBIR on our real dataset. We achieve much higher fidelity with less baking after PBIR. The result is consistent with the quantitative results on synthetic dataset (refer to "Ours - Distilled only" versus "Ours - Full") in 1.

Importance of GI. Our differentiable renderer used in the PBIR stage (Sec. 3.3) is capable of handling global-illumination (GI) effects such as interrelfection. To demonstrate the importance of GI, we run PBIR optimizations with and without GI and compare the material reconstruction qualities. As shown in rows labeled as "Ours - w/o GI" and "Ours - w/o shape ref." in Tab. 1, enabling GI improves the accuracy of material and lighting reconstructions.

A main reason for the usefulness of GI is that, without GI, inverse-rendering optimizations tend to "bake" effects like interreflections into reflectance (*e.g.* albdeo) maps, limiting their overall accuracy. We demonstrate this by comparing albedo maps optimized with and without GI in Fig. 8 (using the balloons data from the MII dataset).

Shape refinement. Lastly, we demonstrate the usefulness of shape refinement (by optimizing the vertex positions of the extracted mesh) using rows labeled as "Ours - w/o shape ref." and "Ours - Full" in Tab. 1 and Fig. 9. Please refer to the supplement for more examples.

5. Discussion and Conclusion

Limitations. Due to the fundamentally under-constrained nature of inverse rendering, our technique is not immunue to "baking" artifacts—especially when predictions provided by our neural stages are far from the groundtruth. Also, although our technique handles global-illumination effects including interreflection, it assumes for opaque materials and does not currently support the reconstruction of transparent or translucent objects.

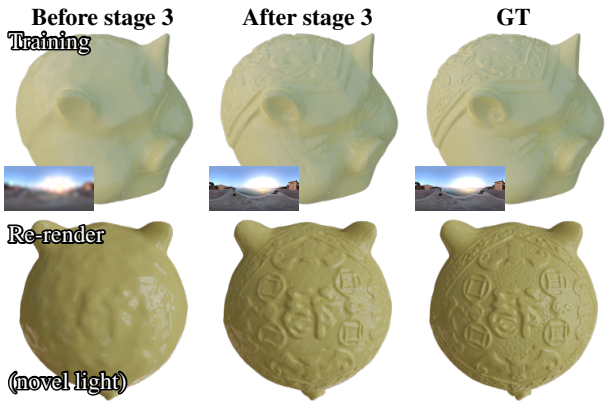


Figure 9: **An example showcasing the usefulness of our PBIR shape refinement.** Our stage 1 can sometimes lack geometry details while our PBIR in stage 3 can recover these details.

Conclusion. We introduced a new method for reconstructing an object's shape and reflectance under unknown environmental illumination. Our technique is comprised of three stages: (i) a surface reconstruction stage fitting an implicit surface and a radiance field from input images; (ii) a neural distilling stage decomposing material and lighting from the radiance field; and (iii) a PBIR stage jointly refining the geometry, material, and lighting. Broad experiments show our effectiveness.

Acknowledgement. This project has been partially supported by NSF grant 1900927.

References

[1] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2447–2456, 2019. 2

[2] Sai Bangaru, Michael Gharbi, Tzu-Mao Li, Fujun Luan, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. Differentiable rendering of neural sdfs through reparameterization. In *ACM SIGGRAPH Asia 2022 Conference Proceedings*, 2022. 5

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, 2021. 1, 2, 3

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P. A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *NeurIPS*, 2021. 1, 2, 3

[5] Brent Burley and Walt Disney Animation Studios. Physically-based shading at Disney. In *Acm Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012. 3

[6] Guangyan Cai, Kai Yan, Zhao Dong, Ioannis Gkioulekas, and Shuang Zhao. Physics-based inverse rendering using combined implicit and explicit geometries. In *Computer*

Graphics Forum, volume 41, pages 129–138. Wiley Online Library, 2022. 2, 3

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 1, 2

[8] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. In *NeurIPS*, 2022. 2

[9] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 1

[10] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *CVPR*, 2022. 2

[11] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. In *NeurIPS*, 2022. 1, 2, 3, 5, 6

[12] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr.jit: A just-in-time compiler for differentiable rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4), July 2022. 2, 5

[13] Rasmus Ramsbøl Jensen, Anders Lindbjerg Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 8

[14] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 4

[15] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[16] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified shape and svbrdf recovery using differentiable monte carlo rendering. In *Computer Graphics Forum*, volume 40, pages 101–113. Wiley Online Library, 2021. 2, 3

[17] Julian Meder and Beat D. Brüderlin. Hemispherical gaussians for accurate light integration. In *ICCVG*, 2018. 4

[18] 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. 1, 2

[19] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 2022. 1, 2

[20] Jacob Munkberg, Wenzheng Chen, Jon Hasselgren, Alex Evans, Tianchang Shen, Thomas Müller, Jun Gao, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022. 1, 2, 3, 4

[21] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 40(6), Dec. 2021. 5

[22] Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. On optimal, minimal brdf sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG)*, 34(6):1–11, 2015. 1

[23] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering. 2021. 2

[24] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 1

[25] Gyutae Park, Sungjoon Son, Jaeyoung Yoo, Seho Kim, and Nojun Kwak. Matteformer: Transformer-based image matting via prior-tokens. In *CVPR*, 2022. 4

[26] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[27] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[28] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 1, 2, 3, 4

[29] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Differentiable signed distance function rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4):125:1–125:18, July 2022. 5

[30] Jiaping Wang, Peiran Ren, Minmin Gong, John M. Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.*, 2009. 4

[31] 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. In *NeurIPS*, 2021. 1, 2, 3, 8

[32] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. *CoRR*, abs/2212.05231, 2022. 8

[33] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. *CoRR*, abs/2208.12697, 2022. 8

[34] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal brdf sampling for two-shot near-field reflectance acquisition. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 1

[35] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 1, 2

[36] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[37] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 2

[38] Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM Trans. Graph.*, 39(4):143:1–143:19, 2020. 2, 5

[39] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *CVPR*, 2022. 2

[40] 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 *CVPR*, 2021. 1, 2, 3, 4

[41] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 2021. 1, 2, 3, 4

[42] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6

[43] Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. Sparse-as-possible svbrdf acquisition. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 1

[44] Jiejie Zhu, Liang Wang, Ruigang Yang, James E Davis, et al. Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1400–1414, 2010. 1