NeMF: Inverse Volume Rendering with Neural Microflake Field

Youjia Zhang[1] Teng Xu[1] Junqing Yu[1] Yuteng Ye[1]
Junle Wang[2] Yanqing Jing[2] Jingyi Yu[3] Wei Yang[1*]
[1]Huazhong University of Science and Technology [2]Tencent [3]ShanghaiTech University

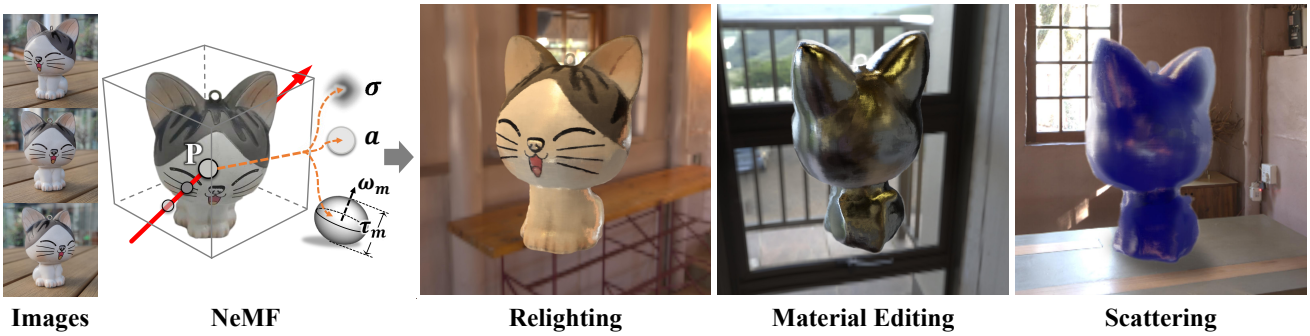

Figure 1. We present the Neural Microflake Field (NeMF) for inverse volumetric rendering from multi-view images under unknown natural illumination. NeMF represents the scene as a microflake volume, in where light reflects or scatters at each spatial location according to volume density, microflake roughness and normal. The optimized NeMF enables high-quality relighting, material editing, synthesize volume scattering effects (shadow is imposed using the preset shadow effect in Microsoft PowerPoint for better exhibition) and etc.

Abstract

Recovering the physical attributes of an object's appearance from its images captured under an unknown illumination is challenging yet essential for photo-realistic rendering. Recent approaches adopt the emerging implicit scene representations and have shown impressive results.However, they unanimously adopt a surface-based representation,and hence can not well handle scenes with very complex geometry, translucent object and etc.In this paper, we propose to conduct inverse volume rendering, in contrast to surface-based, by representing a scene using microflake volume, which assumes the space is filled with infinite small flakes and light reflects or scatters at each spatial location according to microflake distributions. We further adopt the coordinate networks to implicitly encode the microflake volume, and develop a differentiable microflake volume renderer to train the network in an end-to-end way in principle.Our NeMF enables effective recovery of appearance attributes for highly complex geometry and scattering object, enables high-quality relighting, material editing, and especially simulates volume rendering effects, such as scattering, which is infeasible for surface-based approaches.

1. Introduction

Inverse rendering refers to the process of recovering an object's physical attributes related to its appearances, including shape, reflectance and illumination, from its image observations. The above physical attributes play a vital role in graphic applications that require physically reasonable realism. The problem is highly ill-posed due to the complication of object geometries, material and varieties of illuminations. It becomes even more difficult when the images are captured under an unknown illumination condition. The typical practice is to represent the object as surfaces and then solve for the Spatially-Varying Bidirectional Reflectance Distribution Functions (SVBRDF) at each ray-surface interaction. Traditional approaches rely on restrictive assumptions [2, 8, 25, 26, 28] or sophisticated capture systems, such as light-stages [13], co-located flashlight and camera setup [56], and etc.

More recent works explore the implicit scene representations, e.g., radiance field and signed distance functions [24, 32, 40, 52, 62], and achieve promising results. They exploit the geometry, reflectance, or visibility recovered by implicit models as initial estimates for solving the

*Corresponding author: Wei Yang.

ill-posed inverse rendering problem. Notably, NeRD [7] adopts a two-stage estimation strategy by first predicting the sampling pattern and albedo, and then performing per-ray SVBRDF decomposition. NeRFactor [60] applies hard surface approximation on NeRF geometry and recovers neural fields of surface normals, light visibility, albedo and SVBRDFs. Zhang et al. [61] represent the scene geometry as a zero-level set and recover spatially-varying indirect illumination for more accurate inverse rendering. Nevertheless, they either rely on a surface-based representation or need to extract geometry from a volume representation first for the subsequential reflectance estimation. This usually leads to a multi-stage refinement framework, and the performance depends heavily on qualities of the recovered geometries.

Recall that the seminal work of NeRF [34] adopts a radiance volume representation and enables photorealistic rendering without explicit geometry modeling. In essence, NeRF assumes the space is filled with infinite small particles that emit radiation. In this paper, we faithfully extend the volumetric setup by replacing the particles with oriented flakes, which reflect or scatter light according to space occupancies and materials [22].The interaction of light with a collection of microflakes in the volume is described by the microflake phase function, which is determined by the ellipsoidal distribution of normals (NDF) and further parameterized by the microflake normal direction ω_m of and roughness τ_m, as shown in Fig. 1. Such representation can also simulate surface-like object with denser flakes inside the object, while sparser outside, as shown in Fig. 2.With this Neural Microflake Field (NeMF) representation, we propose to conduct inverse volume rendering, to tackle the overly dependency on geometry problem of existing methods.We start from the vanilla NeRF model, add one additional MLP branch for estimating the mircoflake normal ω_m. As for the microflake roughness τ_m, we use a U-shaped MLP network to first encode the material to a feature vector, apply sparsity constraint and then decode it back into albedo a and roughness τ_m. To estimate the density, albedo, and microflake distributions, we develop a differentiable microflake volume renderer and use the photometric loss between the rendered and groundtruth images for supervision. We evaluate the proposed method on both synthetic and real datasets. The experimental results show that our approach outperforms existing methods in terms of rendering quality, and is able to recover scenes with complex geometry and translucent objects. Moreover, our NeMF not only enables effective relighting and material editing but also allows for simulating volume scattering, as shown in Fig. 1.

2. Related Work

Our work is closely related to research in inverse rendering and implicit scene representations.

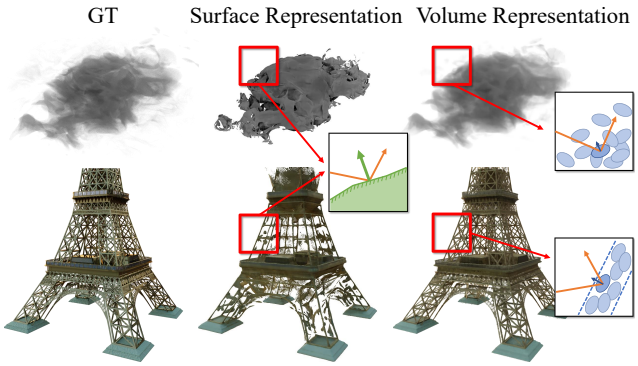

Figure 2. A surface-based representation can not handle scattering materials (e.g., cloud) and very complex geometry (e.g., Eiffel Tower). In contrast, the microflake volume can both handle a surface-like behavior by applying higher density inside the object and low density outside, and a volumetric object.

Inverse rendering. Inverse rendering is a vital problem in both computer vision and computer graphics. One popular way to resolve inverse rendering problems is to use strong scene priors [3, 7, 29–31, 43, 46, 50, 54, 60]. This type of approach recovers intrinsic image properties to infer objects under novel views or illuminations. Another class of inverse rendering methods heavily depends on additional observations. Some of them require input images with known camera parameters [24, 32, 40, 52, 62] or under known lighting source [4–6, 36, 44], while others take 3D geometry obtained from active scanning [21, 27, 41, 44, 59], proxy models [11, 14, 16, 17], silhouette masks [19, 39, 51], or multi-view [20, 36, 42] stereo as a precondition.

Recent work extends inverse rendering to more flexible scene conditions through implicit neural representation and achieves promising results. As exemplary, NeRFactor [60] and PhySG [57] decompose scenes into geometry (NeRFactor extract geometry from NeRF and PhySG relies on SDFs), material and lighting under unknown illumination. PhySG only handles static illumination. NeRFactor models spatially-varying reflectance with low-frequency BRDFs. NeRV [48] considers indirect illumination with known direct illumination. InvRender [61] recovers indirect illumination based on geometry recovered by SDF methods. Most existing approaches either use surface-based representations, e.g., SDF, or extract geometry from volume representations, such as NeRF. Our approach fully relies on volume representation, and hence do not need to recover geometry explicitly. The most related approach to ours is NeRD [7], which calculates the density and SVBRDF parameters for each scene point. However, it accumulates the SVBRDF parameters along a ray as the final material parameter of the ray/pixel, resembling a ray-based reflectance

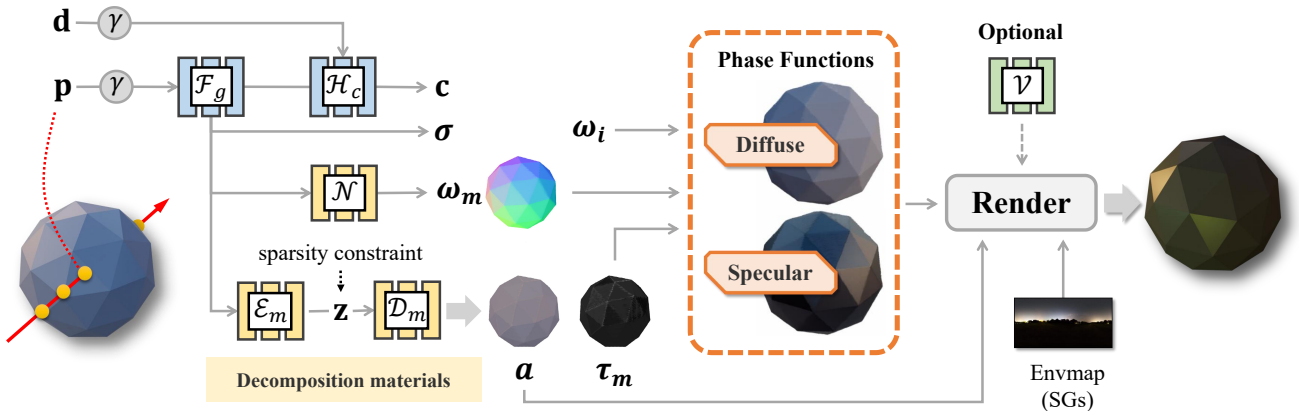

Figure 3. The framework of our NeMF, where we use multiple branches of MLP networks to predict the volume density σ, albedo a, microflake normal ω_m and roughness τ_m respectively. The predicted volume parameters are then sent to a differentiable microflake volume render for rendering.

decomposition, which is fundamentally different from our approach. Different from the prevalence of surface-based inverse rendering problems, research on inverse volume rendering is limited. Existing inverse volume rendering approaches either focus on recovering translucent materials using special acquisition device [18]. or efficiently differentiating the radiative transfer equation [38, 49, 55].

Implicit neural representation. Implicit neural representation is a recent trend that encodes the scene implicitly via a neural network. Typically, they adopt a coordinate network to infer properties related to scene geometry per scene point. NeRF [34] achieves particularly satisfactory performance, enabling photo-realistic novel view synthesis by using MLPs to represent scenes as radiance fields. While NeRF achieves scene representation based on volume, DeepSDF [40, 47] proposes to use a coordinate networks to ecode the zero levelset surface. Occupancy Networks [33] predict the occupancy of each scene point via a coordinate network, the geometry surface then is the place where occupied and vacant exchange. Though these representations both work well for novel view synthesis, they do not model the light transport or reflection in the volume, and hence infeasible for relighting tasks.

Volume Rendering Volume rendering accumulates radiance along the ray passing the volume [15, 37]. Analogous to the microfacet in modeling surfaces, Jakob et al. [23] propose the 'microflake theory' to simulate volumetric scattering with arbitrary microstructures. Extended from the microflake framework, the Symmetric GGX (SGGX) [22] provides the ability to represent microstructure with lightweight storage as well as to represent specular and diffuse microflakes in a unified manner.

3. Neural Microflake Field

In this section, we describe our Neural Microflake Field (NeMF) which introduces the microflake volume model to implicit scene representations.

3.1. The Microflake Distribution

The microflake distribution is designed for modeling spatially-varying properties using oriented non-spherical flakes for simulating light transportation inside a volume. We follow the theory proposed by Heitz [22]. The interaction of light with a collection of microflakes is described by a phase function determined by their distribution of normals (NDF). The NDF serves as a weighting function to scale the radiation transportation in directions. More specifically, the theory simulates the NDF of a collection of microflakes using an ellipsoid, with the ellipsoid's normal w_m and projected areas τ_m onto normal's orthogonal tangent directions ω_x and ω_y. Hence the ellipsoid can be parameterized by w_m and τ_m according to a 3×3 symmetric positive definite matrix S:

$$S = (\omega_x, \omega_y, \omega_m) \begin{pmatrix} \tau_m^2 & 0 & 0 \\ 0 & \tau_m^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} (\omega_x, \omega_y, \omega_m)^T \quad (1)$$

We can model the probability of normals in any given direction ω as on the ellipsoid surface:

$$D(\omega) = \frac{1}{\pi \sqrt{|S|} (\omega^T S^{-1} \omega)^2} \quad (2)$$

With the microflake distributions defined by $D(\cdot)$, we can model the appearance of diffuse materials and specular materials through phase functions, which measure the attenuation given an input light direction ω_l and viewing direction ω_i.

Diffuse Phase Function The phase function of diffuse microflakes then is the integral of attenuation according to angles between normals and incoming and outgoing light directions:

$$f_p^d(\omega_i, \omega_l) = \frac{1}{\pi\tau(\omega_i)} \int_\Omega \langle \omega_l, \omega \rangle \langle \omega_i, \omega \rangle D(w) d_w \quad (3)$$

where $\langle \cdot \rangle$ denotes the dot product, Ω is the hemishpere centered at ω_m, and $\tau(\omega)$ is the projected size of the ellipsoid along direction ω.

Specular Phase Function The phase function for a specular microflakes then is only related to the half angle ω_h as:

$$f_p^s(\omega_i, \omega_l) = \frac{D(\omega_h)}{4\tau(\omega_i)} \quad (4)$$

Notice Eqn. 3 and Eqn. 4 define the reflectance at a certain position in space. The volumetric rendering process requires the integration of light reflected by microflakes along the target ray, and we will explain the process in Sec. 3.3.

3.2. Implicit Microflake Field

We can consider the radiance field representation in NeRF as a volume of particles with view-dependent radiance. Such representation enables photorealistic view synthesis while prohibiting inverse rendering as it's incapable to model light transportation. We find the microflake model can serve as the natural extension of the radiance field for radiation transportation modeling. Recall in the previous section, we can use the microflake normal ω_m and projected area as roughness τ_m to fully represent the microflake distributions. And we adopt a coordinate-based neural network Φ to estimate parameters related to microflake models for each scene point $\mathbf{p}$, as:

$$\Phi : \mathbf{p} \rightarrow (\sigma, a, \omega_m, \tau_m) \quad (5)$$

where σ is the volume density and a is the albedo.

However, using a single network to regress all parameters is impractical. We observe that σ and ω_m are related to scene geometry, and the NeRF structure can provide good estimates. While the albedo a and τ_m correspond to object appearance, we can enforce sparsity constraints according to [61]. Hence we start with the NeRF network $\mathcal{F}_g$ and $\mathcal{H}_c$, add an additional branch $\mathcal{N}$ for estimating the microflake normal, and we use a U-shaped MLP structure with encoder $\mathcal{E}_m$ and decoder $\mathcal{D}_m$ for mapping appearance into sparse latents $\mathbf{z}$. We have:

$$\Phi = \{\mathcal{F}_g, \mathcal{H}_c, \mathcal{N}, \mathcal{E}_m, \mathcal{D}_m\} \quad (6)$$

where:

$$\mathcal{F}_g : \mathbf{p} \rightarrow \sigma;\ \mathcal{F}_g + \mathcal{H}_c : (\mathbf{p}, \mathbf{d}) \rightarrow \mathbf{c};\ \mathcal{F}_g + \mathcal{N} : \mathbf{p} \rightarrow \omega_m$$
$$\mathcal{F}_g + \mathcal{E}_m : \mathbf{p} \rightarrow \mathbf{z};\ \mathcal{D}_m : \mathbf{z} \rightarrow (a, \tau_m) \quad (7)$$

$\mathbf{d}$ is the query ray direction. $\mathbf{c}$ is the color. The implicate networks used for modeling the microflake field is shown in Fig. 3.

3.3. Rendering with NeMF

Our NeMF represents the scene as a distribution of microflakes at every point in space. Rendering with the microflake volume follows the principles provide in [22], i.e, for a given ray $\mathbf{r}$ with direction $\mathbf{d}$ passing through the volume, the resulting pixel intensity is integral of radiance along $\mathbf{r}$.

$$C(\mathbf{r}) = \int_{t_n}^{t_f} \eta(t)\sigma(\mathbf{r}_t)\nu(\mathbf{r}_t, \omega_i) d_t \quad (8)$$

where $\mathbf{r}_t$ means the point on $\mathbf{r}$ at t, $\eta(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}_s) d_s)$ is the weight, $\nu(\mathbf{r}_t, \omega_i)$ is the radiance at point $\mathbf{r}_t$ in the direction ω_i of $\mathbf{r}$, where $\omega_i = -\mathbf{d}$ is a unit vector pointing from a point in space to the camera. For NeRF, the radiance is the view-dependent color to be predicted by the network. In our NeMF, the radiance at a scene point x refers to its transported radiation, and is an integral of all incoming light that reaches at x, attenuated by the phase function given outgoing light direction (i.e., the viewing direction $\mathbf{d}$). Hence, we can calculate $\nu(\mathbf{r}_t, \omega_i)$ with a known phase function f_p as:

$$\nu(\mathbf{r}_t, \omega_i, f_p) = \alpha \int_\Omega f_p[\omega_i, \omega_m(\mathbf{r}_t), \omega_l] \cdot L(\mathbf{r}_t, \omega_l) d_{\omega_l} \quad (9)$$

where $L(\mathbf{r}_t, \omega_l)$ is the light intensity reaches $\mathbf{r}_t$ in direction ω_l. Recall that we have separated the microflake phase functions into diffuse and specular components, hence we have the final formula for ν as:

$$\nu^*(\mathbf{r}_t, \omega_i) = \nu(\mathbf{r}_t, \omega_i, f_p^d) + \nu(\mathbf{r}_t, \omega_i, f_p^s) \quad (10)$$

Combining Eqn. 3, 4, 10, 9, and 8, we obtain the rendering equations of our NeMF. However, in practice we have to adapt the discrete data from sampling, we replace integral with summation over discrete sampling positions. Then another issue to be mindful is the phase functions measure light attenuates according to incoming and outgoing directions. When we calculate $\nu^*(\cdot)$, we need to conduct importance ray sampling of light directions, which is elaborated as Algorithm 1 in Heitz et al. [22] .

4. Inverse Volumetric Rendering

Now that we have the network structure of NeMF, and regression of the density, abledo, microflake normal, and roughness for each position is inverse volume rendering. We have provided the volumetric rendering functions for NeMF in Sec. 3.3, and it is differentiable. We can render from NeMF and use the photometric loss between the

rendering result and image observations as supervision for training, i.e.:

$$\mathcal{L}_c = \sum_{\mathbf{r},I} \|C(\mathbf{r}) - I(\mathbf{r})\|_2^2 \tag{11}$$

where $I(\mathbf{r})$ is the color of ray $\mathbf{r}$ in image I. However, only using the photometric loss is not sufficient for producing hight quality results. We add the density field loss $\mathcal{L}_\sigma = \sum_k \eta_k(\|\omega_m - \Delta\sigma\|_2^2 + \max(0, \langle\omega_m, \omega_i\rangle))$ presented in RefNeRF [60], the latent sparsity $\mathcal{L}_z = \sum_{j=1}^{n} \mathrm{KL}(\rho\|\hat{\rho}_j)$ ($\hat{\rho}_j$ is the average the j^{th} channel of $\mathbf{z}$ over batch input, ρ is set to 0.05) as in InvRender [61], for regularization.

We further enforce smoothness on both microflake normals and latent $\mathbf{z}$ using the following smoothness loss:

$$\begin{aligned}\mathcal{L}_s = &\|\mathcal{N}(\mathcal{F}_g(\mathbf{p})) - \mathcal{N}(\mathcal{F}_g(\mathbf{p}+\epsilon))\|_1 \\ &+\|\mathcal{D}_m(\mathbf{z}) - \mathcal{D}_m(\mathbf{z}+\epsilon)\|_1\end{aligned} \tag{12}$$

where ϵ is a small random variable drawn from a Gaussian distribution with a mean of zero and a variance of 0.01.

The final loss is the sum of all previously defined losses:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_z + \mathcal{L}_\sigma + \mathcal{L}_s \tag{13}$$

where the corresponding weights of each term are ignored for clear presentation.

4.1. Illumination and Visibility

During training, the illumination is unknown and we need to recover simultaneously with the microflake field. We assume that all lights come from an infinitely faraway background and parameterize them as 128-dimensional Spherical Gaussians (SGs), a common practice in inverse rendering. To further improve the quality, we model the visibility of direct illumination for each scene point following InvRender [61] and Relighting4D [12]. Specifically, we compute the visibility of each point w.r.t. the light direction ω_l by marching a ray from the scene to the light in the background and calculating its opacity η (Eqn. 8) with a pre-trained NeRF. We further encode the visibility of environment light w.r.t. a scene point using the SG parameterization, and use an MLP network $\mathcal{V}$ to overfit the visibility SG parameters for each spatial position. Recall that the radiance at a point x, $\nu^*(x, \omega_i)$ contains a diffuse component $\nu(x, \omega_i, f_p^d)$ and a specular component $\nu(x, \omega_i, f_p^s)$. Inspired by previous work [57, 61], we use the summation of multiplication of environment lighting and visibility as the diffuse component for faster computation:

$$\begin{aligned}\nu_d(x) = \frac{a}{\pi} \sum_{\omega_l \in \Omega} \big\{&[V(x, \omega_l) Y(\omega_l)] \otimes \\ &[L^{\mathrm{sg}}(\omega_l) Y(\omega_l)] \cdot \langle \omega_l \cdot \omega_m \rangle\big\}\end{aligned} \tag{14}$$

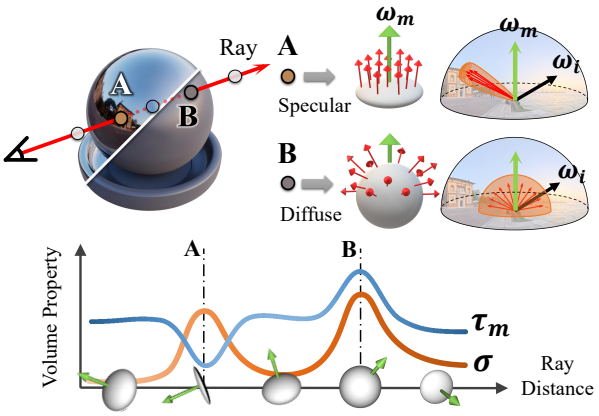

Figure 4. The rendering process of a ray: point A is relatively specular and point B is relatively diffuse. The ellipsoid that simulates the microflake distribution of A is thiner and that of B is rounder. The final color of ray is integral of transported radiation from the environment light by microflakes along the ray.

where $Y(\omega_l)$ is the SG basis. For the specular reflectance $\nu(x, \omega_i, f_p^s)$, we check the visibility of point ω_l w.r.t. the light direction during the ray sampling, and set it to 0 if the light ω_l is invisible to x. We use the visibility adapted color as our final rendered color.

With an optimized NeMF, we can relight the captured object by changing the illumination SGs in our microflake volume renderer, edit the material by mapping the albedo to a new color, and create novel volume rendering effects (such as scattering) through changing the weighting factor between diffuse and specular radiance in Eqn. 10 .

5. Experiments

We conduct experiments on both synthetic and real-world datasets to evaluate our proposed NeMF.

5.1. Synthetic Data

We use 4 synthetic Blender scenes (balloon, mic, spot, polyhedron) to validate our model. For each object, we choose a specific natural illumination. We render 100 training images for each object under the selected illumination via Blender Cycles, and disable all non-standard post-rendering effects. We also render 200 test images, along with their albedo maps and relighting images using other two environment maps to evaluate the novel view synthesis and relighting performance of our model. All images are in the PNG format and the image resolution is 800x800.

5.2. Ablation Study

We validate our design choices by ablating 4 major model variants, that are without density and mircoflake nor-

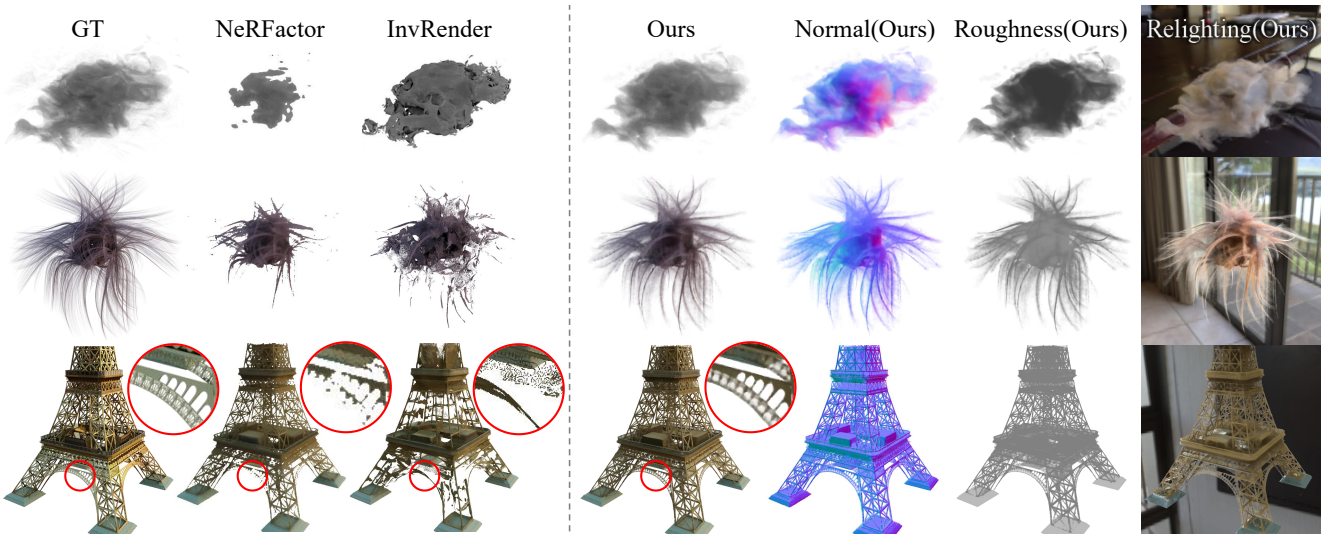

Figure 5. **Comparison on objects with complex geometries** We compare our NeMF with NeRFactor [60] and InvRender [61] for synthetic objects with very complicated geometry and material, including a cloud (top), a furry ball (middle), and Eiffel Tower (bottom).

Method	Roughness	Albedo			View Synthesis			Relighting		
	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PhySG [57]	-	15.3543	0.8811	0.2463	15.1026	0.8883	0.2399	14.5316	0.8804	0.2440
NeRFactor [60]	-	20.3366	0.9193	0.1093	23.2638	0.9429	0.0997	21.3415	0.9254	0.1056
InvRender [61]	0.0121	21.9568	0.9339	**0.0811**	27.4874	0.9639	0.0816	24.5309	0.9555	0.0897
Ours	**0.0074**	**23.6982**	**0.9387**	0.0870	**28.8920**	**0.9678**	**0.0744**	**25.7048**	**0.9561**	**0.0864**
w/o normal	0.0672	15.4516	0.8914	0.1359	28.1014	0.9627	0.0899	18.6333	0.9134	0.1283
w/o vis.	0.0445	19.2018	0.9231	0.1057	28.6739	0.9652	0.0834	20.5587	0.9091	0.1339
w/o smoothness	0.0075	23.5721	0.9360	0.0906	28.8234	0.9638	0.0874	24.9586	0.9445	0.1081
w/o latent space	0.0160	23.2460	0.9407	0.0930	28.8529	0.9670	0.0775	24.8025	0.9534	0.0945
fewer samples	0.0527	19.9179	0.9128	0.1099	28.8310	0.9672	0.0771	22.2603	0.9301	0.1170

Table 1. **Quantitative evaluations.** We present the average results on the test images of all four synthetic scenes. Though InvRender achieves slightly better LPIPS on albedo estimation, our full model achieves the best performance on roughness estimation, view synthesis and relighting.

mal regularizations, without visibility modeling, without sparsity constraints on the latent code, and without smoothness regularizations. We compare them with our NeMF model to observe whether there is a performance drop quantitatively or qualitatively. We present our quantitative ablation studies in Tab. 1, and the qualitative ablation studies in Fig. 7.

Fig. 7 shows that "w/o normal" bakes the ambient lighting into the surface of objects, generating the worst results among 4 ablation settings. In "w/o vis.", we train a model without calculating the visibility of each location point to constrain our model during pre-training. "w/o vis." assumes the environment illumination is visible to all scene points without considering self-occlusion and interreflection. This setup leads to incorrect predictions, especially around the occlusion boundaries. "w/o latent space" maps location points to parameters of neural network straightforwardly, instead of transforming it to a latent space first. This results in incorrect extraction of the object's materials from images, as well as vaguer edges compared to our model.

5.3. Comparison

We compare our NeMF with 3 baselines (NeRFactor [60], PhySG [57] and InvRender [61]) in the tasks of novel view synthesis and relighting on the synthetic datasets. We use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as metrics. We also use Learned Perceptual Image Patch Similarity

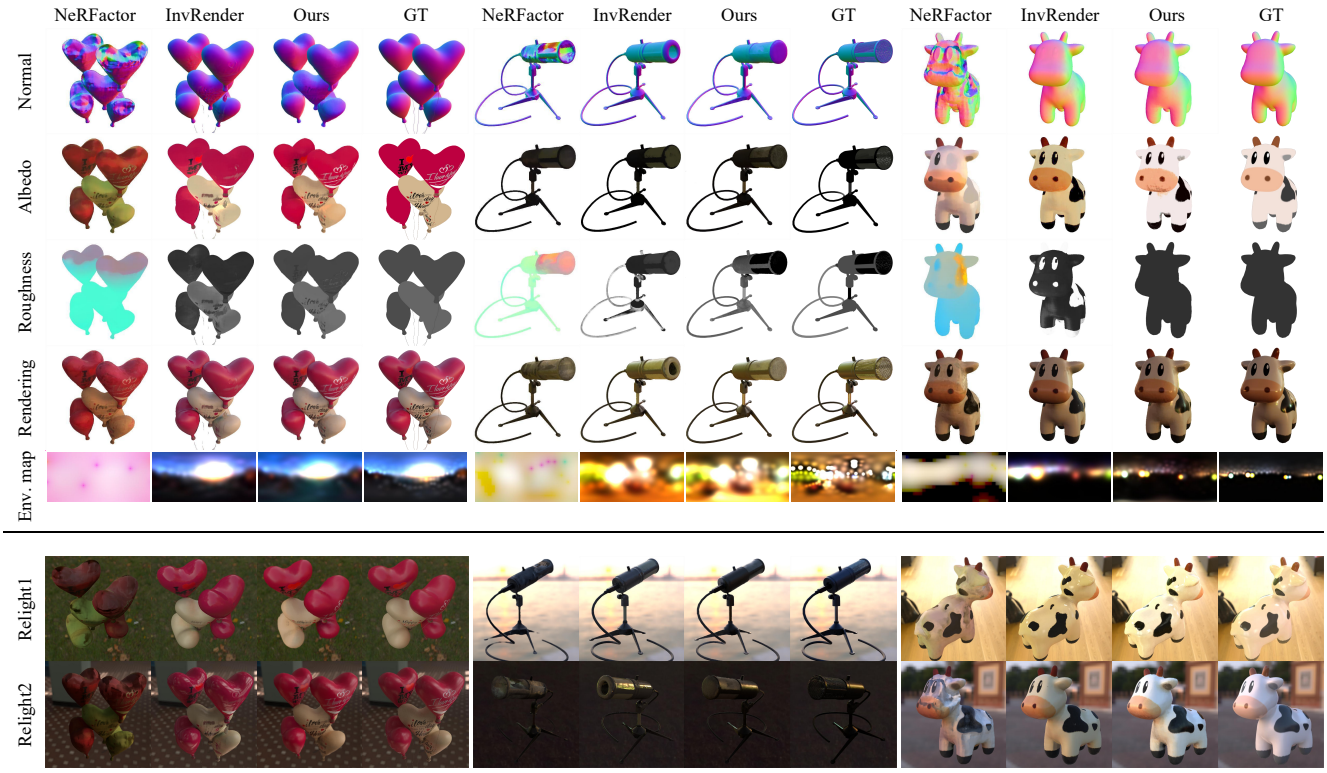

Figure 6. **Comparisons with other approaches.** We show the normal, albedo, roughness, and environment map estimated by NeRFactor, InvRender and our method on three scenes. We also compared the rendering results of the new viewpoint under three different lighting conditions (the original light and two novel lights). Note that the roughness of NeRFactor is visualized with the latent code.

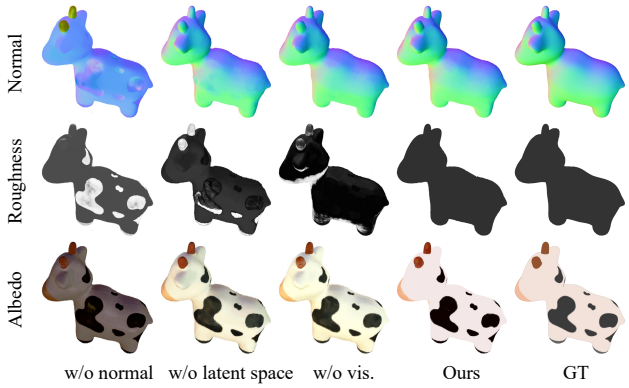

Figure 7. **Ablation study on a synthetic scene (Sec. 5.2).**

(LPIPS) [58], where lower is better. Tab. 1 demonstrates that there is a bigger gap between PhySG's results and ground truth. The main reason is that PhySG assumes the object recovered is homogeneous, therefore it cannot even recover the shapes of objects with multiple materials, the mic for example.Tab. 1 and Fig. 6 show that our model has a better performance both quantitatively and qualitatively than NeRFactor. Fig. 6 demonstrates that NeRFactor cannot extract an accurate environment map, and has a lower performance on the balloon datasets especially, as the albedo is baked into the environment map and thus leads to poor relighting results. Compared to NeRFactor, InvRender and our model have much smoother results on normal estimation, which are closer to ground truth.As shown in Fig. 6 and Tab. 1, although InvRender has more access to ground truth on both novel view synthesis and relighting than previous work, it fails to recover some details of scenes compared to our model. The results also demonstrate that InvRender may not be able to decompose the scene into geometry and illumination accurately when the surface of a scene is complex.

5.4. Real Data Experiments

To demonstrate that our approach is able to handle real-work data, we test on 5 datasets including the 'cat', 'monk', 'teapot' as concrete and simple objects, and two complicate scenes, 'furry ball' and 'bands'. We capture all 'cat', 'monk', 'furry ball' and 'bands' scenes with a mobile phone, OPPO RENO5, and record a video while the collector walking around the object. The frames and the

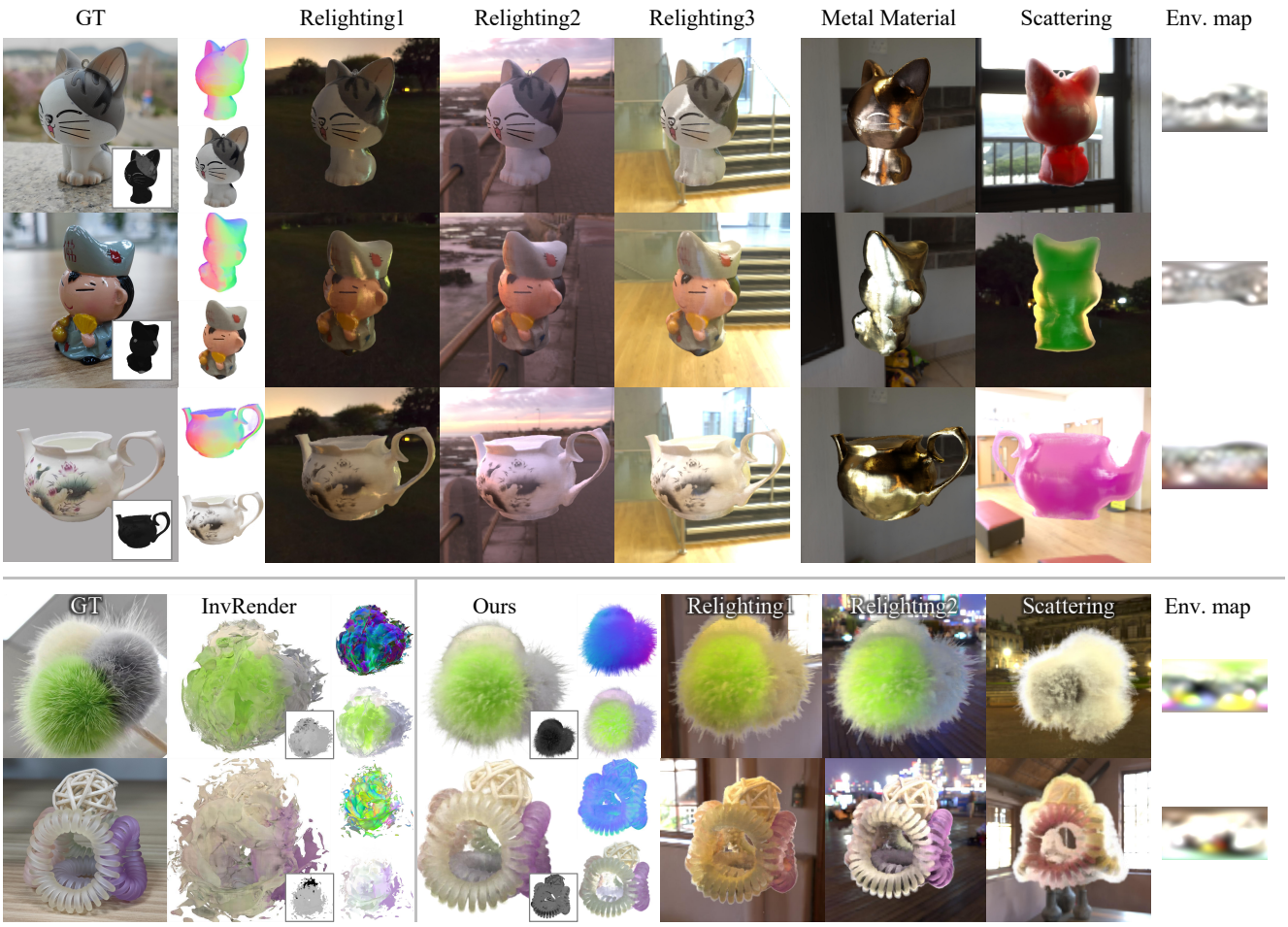

Figure 8. **Results on real captures.** Our method is capable to handle real-world objects composed of multiple materials. Top three rows, we show results on concrete objects. For each scene, we present the groundtruth, appearance properties, relighting under three novel illuminations, material editing, scattering and environment maps.

resolution of each video are 30fps and 960x540, respectively. We extract around 400 frames from the video for each object, and from which we randomly select 100 images for training. We obtain the camera parameters using COLMAP [45] (model: "PINHOLE"). Before training, we remove the background of all images using background removal tools [1]. We illustrate the relighting, material editing and volume scattering results in Fig. 8. As we can see, the relighting result is faithful, given that we capture the data with a mobile phone in office environment lighting. The metal-like effect of the 'cat' and 'monk' is very realistic. While generating volume scattering effect, we change the object color while preserving the specular terms and give the object a semi-translucent appearance, which is infeasible for most surface based inverse rendering approaches. Particularly, our NeMF can handle the complicate objects very well, as shown bottom two rows in Fig. 8.

6. Conclusion

In this paper, we present the Neural Microflake Field (NeMF), which uses an implicit coordinate network to encode a microflake volume for inverse volumetric rendering. Our NeMF is a fully volumetric model and can well handle complicate geometries and materials. Our NeMF still has several limitations: the recovered geometry is not as smooth as the neural surface-based methods, we adopt a per-scene optimization scheme for training which is not generalizable, our approach requires a large number of input images (over 100), and both training and rendering are time-intensive. In the future, we plan to address above issues by incorporating recent advances, including the multi-resolution hashing technique in InstantNGP [35], and factorization methods [9, 10]. We would like to further adapt our trained network to parameterizations (PlenOctree [53]) for real-time rendering and relighting.

References

[1] 1. Natural population growth rate. https://www.remove.bg/. Accessed: October 1, 2022. 8

[2] Louis-Philippe Asselin, Denis Laurendeau, and Jean-Francois Lalonde. Deep svbrdf estimation on real materials. In *2020 International Conference on 3D Vision (3DV)*, pages 1157–1166. IEEE, 2020. 1

[3] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2014. 2

[4] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[5] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *European Conference on Computer Vision*, pages 294–311. Springer, 2020. 2

[6] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020. 2

[7] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2

[8] Mark Boss, Fabian Groh, Sebastian Herholz, and Hendrik PA Lensch. Deep dual loss brdf parameter estimation. In *MAM@ EGSR*, pages 41–44, 2018. 1

[9] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 8

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 8

[11] Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N Kutulakos, and Jingyi Yu. A neural rendering framework for free-viewpoint relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5599–5610, 2020. 2

[12] Zhaoxi Chen and Ziwei Liu. Relighting4d: Neural relightable human from videos. In *European Conference on Computer Vision*, pages 606–623. Springer, 2022. 5

[13] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000. 1

[14] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG)*, 33(6):1–12, 2014. 2

[15] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988. 3

[16] Duan Gao, Guojun Chen, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 2

[17] Stamatios Georgoulis, Vincent Vanweddingen, Marc Proesmans, and Luc Van Gool. A gaussian process latent variable model for brdf inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3559–3567, 2015. 2

[18] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):1–13, 2013. 3

[19] Clement Godard, Peter Hedman, Wenbin Li, and Gabriel J Brostow. Multi-view reconstruction of highly specular surfaces in uncontrolled environments. In *2015 International Conference on 3D Vision*, pages 19–27. IEEE, 2015. 2

[20] Purvi Goel, Loudon Cohen, James Guesman, Vikas Thamizharasan, James Tompkin, and Daniel Ritchie. Shape from tracing: Towards reconstructing 3d object geometry and svbrdf material from images via differentiable path tracing. In *2020 International Conference on 3D Vision (3DV)*, pages 1186–1195. IEEE, 2020. 2

[21] 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. 2

[22] Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. The sggx microflake distribution. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 2, 3, 4

[23] Wenzel Jakob, Adam Arbree, Jonathan T Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. In *ACM SIGGRAPH 2010 papers*, pages 1–13. 2010. 3

[24] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1251–1261, 2020. 1, 2

[25] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient brdf importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 23(3):496–505, 2004. 1

[26] Hendrik Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Work-*

shop on Rendering Techniques, pages 103–114. Springer, 2001. 1

[27] Hendrik PA Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics (TOG)*, 22(2):234–257, 2003. 2

[28] 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

[29] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 2

[30] 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(6):1–11, 2018. 2

[31] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6123–6133, 2021. 2

[32] Robert Maier, Kihwan Kim, Daniel Cremers, Jan Kautz, and Matthias Nießner. Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proceedings of the IEEE international conference on computer vision*, pages 3114–3122, 2017. 1, 2

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 3

[34] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 8

[36] 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(6):1–12, 2018. 2

[37] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 3

[38] Merlin Nimier-David, Thomas Müller, Alexander Keller, and Wenzel Jakob. Unbiased inverse volume rendering with differential trackers. *ACM Transactions on Graphics (TOG)*, 41(4):1–20, 2022. 3

[39] Geoffrey Oxholm and Ko Nishino. Multiview shape and reflectance from natural illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2155–2162, 2014. 2

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1, 2, 3

[41] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1417–1427, 2020. 2

[42] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Trans. Graph.*, 38(4):78–1, 2019. 2

[43] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020. 2

[44] Carolin Schmitt, Simon Donne, Gernot Riegler, Vladlen Koltun, and Andreas Geiger. On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3493–3503, 2020. 2

[45] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 8

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

[47] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020. 3

[48] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[49] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 3

[50] Xin Wei, Guojun Chen, Yue Dong, Stephen Lin, and Xin Tong. Object-based illumination estimation with rendering-aware neural networks. In *European Conference on Computer Vision*, pages 380–396. Springer, 2020. 2

[51] 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(6):1–12, 2016. 2

[52] 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*, 33:2492–2502, 2020. 1, 2

[53] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 8

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

[55] Cheng Zhang, Zihan Yu, and Shuang Zhao. Path-space differentiable rendering of participating media. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021. 3

[56] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022. 1

[57] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2, 5, 6

[58] 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. 7

[59] 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. 2

[60] 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. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2, 5, 6

[61] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022. 2, 4, 5, 6

[62] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015. 1, 2