NeRD: Neural Reflectance Decomposition from Image Collections

Mark Boss[1], Raphael Braun[1], Varun Jampani[2], Jonathan T. Barron[2],
Ce Liu[2], Hendrik P.A. Lensch[1]

[1]University of Tübingen, [2]Google Research

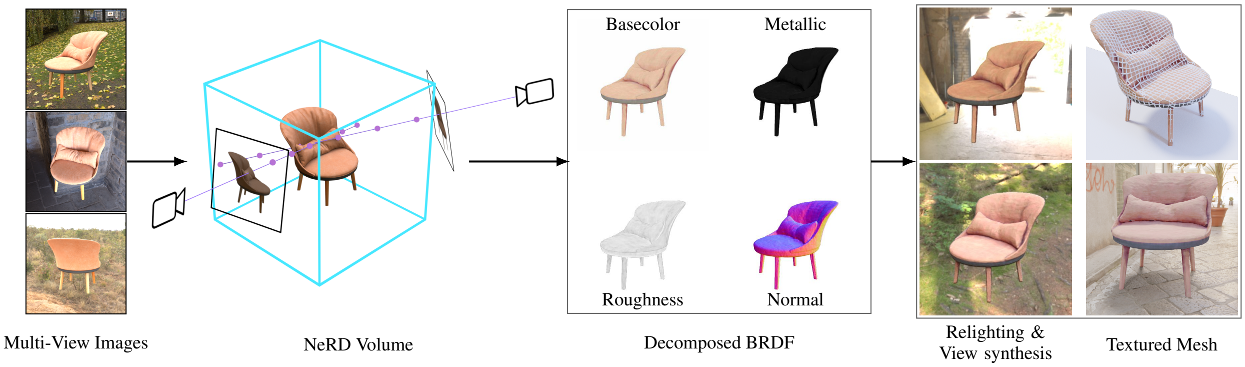


Figure 1: **Neural Reflectance Decomposition for Relighting.** We encode multiple views of an object under varying or fixed illumination into the NeRD volume. We decompose each given image into geometry, spatially-varying BRDF parameters and a rough approximation of the incident illumination in a globally consistent manner. We then extract a relightable textured mesh that can be re-rendered under novel illumination conditions in real-time.

Abstract

Decomposing a scene into its shape, reflectance, and illumination is a challenging but important problem in computer vision and graphics. This problem is inherently more challenging when the illumination is not a single light source under laboratory conditions but is instead an unconstrained environmental illumination. Though recent work has shown that implicit representations can be used to model the radiance field of an object, most of these techniques only enable view synthesis and not relighting. Additionally, evaluating these radiance fields is resource and time-intensive. We propose a neural reflectance decomposition (NeRD) technique that uses physically-based rendering to decompose the scene into spatially varying BRDF material properties. In contrast to existing techniques, our input images can be captured under different illumination conditions. In addition, we also propose techniques to convert the learned reflectance volume into a relightable textured mesh enabling fast real-time rendering with novel illuminations. We demonstrate the potential of the proposed approach with experiments on both synthetic and real datasets, where we are able to obtain high-quality relightable 3D assets from image collections. The datasets and code are available at the project page: `https://markboss.me/publication/2021-nerd/`.

1. Introduction

Capturing the geometry and material properties of an object is essential for several computer vision and graphics applications such as view synthesis [10, 54], relighting [5, 10, 21, 22, 30, 55], object insertion [7, 20, 30] *etc*. This problem is often referred to as *inverse rendering* [24, 41], where shape and material properties are estimated from a set of images, *e.g*., representing the surface properties as spatially-varying Bidirectional Reflectance Distribution functions (SVBRDF) [38].

Modeled according to physics, the reflected color observed by a viewer is the integral of the product of SVBRDF and the incoming illumination over the hemisphere around that surface's normal [23]. Disentangling this integral and estimating shape, illumination, and SVBRDF from images is a highly ill-posed and underconstrained inverse problem. For instance, an image region may appear dark either due to a dark surface color (material), the absence of incident light at that surface (illumination), or due to the normal of that surface facing away from the incident light (shape).

Traditional SVBRDF estimation techniques involve capturing images using a light-stage setup where the light direction and camera view settings are controlled [4, 9, 26, 27, 28]. More recent approaches for SVBRDF estimation employ more practical capture setups [6, 7, 8, 10, 19, 37], but limit the illumination to a single dominant source (*e.g*., a flash attached to a camera). Assuming known illumination or constraining its complexity significantly reduces the ambiguity of shape and material estimation and limits the practical utility to laboratory settings or to flash photography in dark environments.

In contrast to standard SVBRDF and shape estimation techniques, recently introduced coordinate-based scene representation networks such as Neural Radiance Fields (NeRF) [34, 36, 60] can directly perform high-quality view synthesis without explicitly estimating shape or SVBRDF. They represent the radiance field of the scene using a neural network trained specifically for a single scene, using as input multiple images of that scene. These neural networks directly encode the geometry and appearance as volumetric density and color functions parameterized by 3D coordinates of query points in the scene. Realistic novel views can be generated by raymarching through the volume. Though these approaches are capable of reproducing view-dependent appearance effects, the radiance of a point in a direction is "baked in" to these networks, making them unusable for relighting. Even if such techniques could be extended to relighting, the rendering speed of these methods limits their practicality — the time required by NeRF to generate a single view is about 30 seconds [36].

This work presents a shape and SVBRDF estimation technique that allows for a more flexible capture setting while enabling relighting under novel illuminations. Our key technique is an explicit decomposition model for shape, reflectance, and illumination within a NeRF-like coordinate-based neural representation framework [36]. Compared to NeRF, our volumetric geometry representation stores SVBRDF parameters at each 3D point instead of radiance. Each image is then differentiably rendered with a jointly optimized spherical Gaussian illumination model (see Figure 1). Shape, BRDF parameters, and illumination are all optimized simultaneously to minimize the photometric rendering loss w.r.t. each input image. We call our approach "Neural Reflectance Decomposition" (NeRD)

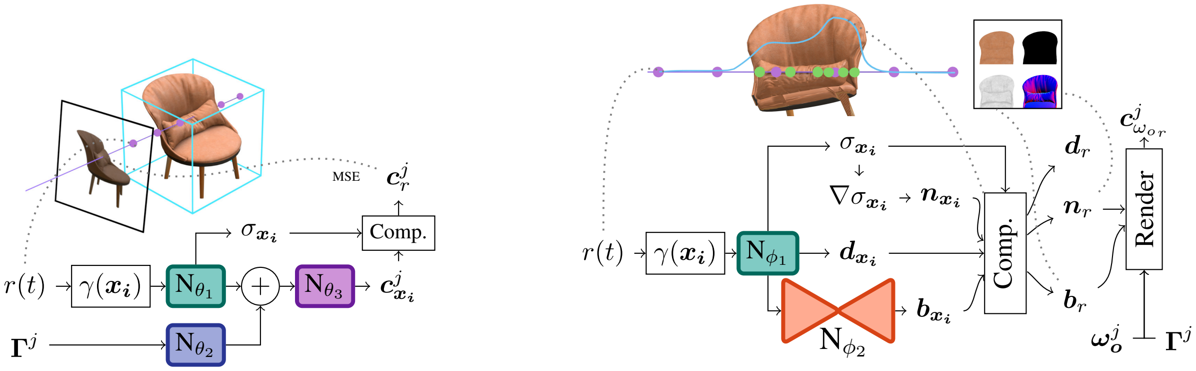
NeRD not only enables simultaneous relighting and view synthesis but also allows for a more flexible range of image acquisition settings: Input images of the object need not be captured under the same illumination conditions. NeRD supports both camera motion around an object as well as captures of rotating objects. All NeRD requires as input is a set of images of an object with known camera pose (computed for *e.g*. using COLMAP [43, 44]), where each image is accompanied by a foreground segmentation mask. Besides the SVBRDF and shape parameters, we also explicitly optimize the illumination corresponding to each image for varying illuminations or globally for static illumination.

As a post-processing step, we propose a way to extract a 3D surface mesh along with SVBRDF parameters as textures from the learned coordinate-based representation network. This allows for a highly flexible representation for downstream tasks such as real-time rendering of novel views, relighting, 3D asset generation, *etc*.

2. Related Work

Neural scene representations. Recently, neural scene representations have attracted considerable attention [33, 34, 35, 36, 46, 47, 53, 60, 56]. These methods surpassed previous state-of-the-art in novel view interpolation and achieved photo-realistic results in most cases. The primary innovation of these methods is to model a scene using a volumetric, voxel or implicit representation, and then train a neural network per object to represent it. Because these neural representations are inherently 3D, they enable novel view synthesis. Our approach follows a similar representation but decomposes the appearance into shape, BRDF and illumination. One significant concern with these approaches is their long training and inference time [36]. We address the latter issue by explicitly extracting a surface mesh and BRDF parameters to make use of the learned 3D model in common game engines or path tracers. Some concurrent works [48, 59] also try to estimate BRDFs in neural volumes. In NeRV [48], the illumination is assumed to be known. Another work, PhySG [59] also leverages spherical Gaussians to model the illumination, but constraints itself to scenes under a fixed illumination, compared to our setup which handles both fixed and varying illumination.

BRDF estimation. Though highly accurate BRDF measurements can be achieved under laboratory conditions with known view and light positions [4, 9, 26, 27, 28], the complicated setup of these methods often renders on-site material capture infeasible. Methods aiming at "casual" capture frequently rely on neural networks to learn a prior on the relationship between images and their underlying BRDFs. Often, planar surfaces under camera flash illumination are considered for single-shot [2, 15, 31, 42], few-shot [2] or multi-shot [3, 9, 16, 17, 19] estimation. This casual setup can be extended to estimating the BRDF and shape of objects [7, 8, 10, 37, 58] or scenes [45]. Recently, Bi *et al*. [6] leveraged a NeRF-style framework to decompose a scene into shape and BRDF parameters with a single co-located light source. Uncontrolled natural illumination adds additional ambiguities, which are partially addressed by self augmented networks [29, 57] that work on single input images. However, their SVBRDF model assumes homogeneous specularities. Full SVBRDF estimations in natural light setups have been proposed by Dong *et al*. [18] by explicitly optimizing for illumination and reflectance from temporal appearance traces of rotating objects. Later, geometry reconstruction was added to the process [52]. Our method supports more flexible and practical capture settings.



(a) **Sampling Network.** The main task of the coarse sampling network is to generate a finer distribution for sampling in the decomposition network. To match the input during training the color prediction needs to account for the illumination. We combine a compacted Γ^j from N_{θ_2} with the latent color output of N_{θ_1} to generate the illumination-dependent color in N_{θ_3}.

(b) **Decomposition Network.** With the sampling pattern generated from the coarse network, we perform SVBRDF decomposition at each point in neural volume. The density, σ, and direct RGB color $\boldsymbol{d}$ is queried from the N_{ϕ_1}. Additionally, a vector is passed to N_{ϕ_2}, which decodes it to the point's BRDF parameters $\boldsymbol{b}$. By compressing the BRDF to a low-dimensional latent space, all surface points contribute to training a joint space of plausible BRDFs for the scene. Each point still interpolates its parameters in this space. The gradient from the density forms the normal $\boldsymbol{n}$ and is passed with the BRDF and spherical Gaussians Γ^j to the differentiable renderer.

Figure 2: **NeRD Architecture.** The architecture consists of two networks. Here, N_{θ_1}/N_{ϕ_1} denote instances of the main network which encodes the reflectance volume. $r(t)$ defines a ray with sampling positions $\boldsymbol{x_i}$, $\gamma(\boldsymbol{x})$ is the Fourier Embedding [36], and $\boldsymbol{\Gamma}^j$ denotes the SG parameters per image j. $\boldsymbol{c}$ is the output color and σ is the density in the volume. The individual samples along the ray need to be alpha composed based on the density σ along the ray. This is denoted as "Comp.".

3. Method

Our method jointly optimizes a model for shape, BRDF, and illumination by minimizing the photometric error to input image collection of an object that are captured under fixed or different illuminations.

Problem setup. Our input consists of a set of q images with s pixels each, $I_j \in \mathbb{R}^{s\times 3}; j \in 1, ..., q$ potentially captured under different illumination conditions. We aim to learn a 3D volume $\mathcal{V}$, where at each point $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$ in 3D space, we estimate BRDF parameters $\boldsymbol{b} \in \mathbb{R}^5$, surface normal $\boldsymbol{n} \in \mathbb{R}^3$ and density $\sigma \in \mathbb{R}$. The environment maps are represented by spherical Gaussian mixtures (SG) with parameters $\boldsymbol{\Gamma} \in \mathbb{R}^{24\times 7}$ (24 lobes).

Preliminaries. We follow the general architecture of NeRF [36]. NeRF creates a neural volume for novel view synthesis using two Multi-Layer-Perceptrons (MLP). NeRF model encodes view-dependent color and object density information at each point in 3D space using MLPs. NeRF consists of two MLPs in which a course *sampling network* samples the volume in a fixed grid and learns the rough shape of an object *i.e.* estimating density σ at a given input 3D location (x, y, z). The second finer network uses this course density information to generate a more dense sampling pattern along the viewing ray where higher density gradients are located. Formally, rays can be defined as $r(t) = \boldsymbol{o} + t\boldsymbol{d}$ with ray origin $\boldsymbol{o}$ and the ray direction $\boldsymbol{d}$. Each ray is cast through the image plane and samples a different pixel location with corresponding color $\hat{\boldsymbol{c}}_r^j$. Marching along the ray through the volume at each sample coordinate $\boldsymbol{x} = (x, y, z)$, the networks $N_{(.)}$ are queried for the volume parameters $\boldsymbol{p}(t)$. Here, we use $\boldsymbol{p}(t)$ as a stand in for the color $\boldsymbol{c}(t)$, density $\sigma(t)$ or in our case BRDF parameters $\boldsymbol{b}(t)$. Following Tancik *et al.* [49] and Mildenhall *et al.* [36], which showed that coordinate-based approaches struggle with learning details based on high frequency $\boldsymbol{x}$ inputs, we also use their proposed Fourier embedding $\gamma(\boldsymbol{x})$ representation of a 3D point. The sampled volume parameters are combined along the ray via alpha composition (Comp.) using the density at each point $\sigma(t)$: $P(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\boldsymbol{p}(t)\,dt$ with $T(t) = \exp\left(-\int_{t_n}^{t}\sigma(s)\,ds\right)$ [36], based on the near and far bounds of the ray t_n and t_f respectively.

NeRD overview. In comparison to NeRF, NeRD architecture mainly differs in the second finer network. NeRD uses *decomposition network* as a finer network which stores the lighting independent reflectance parameters instead of the direct view-dependent color. Also, the sampling network in NeRD differs from NeRF as we learn illumination dependent colors as NeRD can work with differently illuminated input images. An overview of both networks is shown in Fig. 2. The parameters of the networks and the SGs are optimized by backpropagation informed by comparing the output of a differentiable rendering step to each input image

I_j for individual rays across the 3D volume.

Sampling network. The *sampling network* directly estimates a view-independent but illumination dependent color $\boldsymbol{c}^j$ at each point, which is optimized by a MSE: $\frac{1}{s}\sum^{s}(\hat{\boldsymbol{c}}_r^j - \boldsymbol{c}_r^j)^2$. The sampling network's main goal is to establish a useful sampling pattern for the *decomposition network*. The sample network structure is visualised in Fig. 2a. Compared to NeRF, our training images can have varying illuminations. Therefore, the network needs to consider the illumination $\boldsymbol{\Gamma}^j$ to create the illumination dependent color $\boldsymbol{c}^j$ that should match image I_j. The density σ is not dependent on the illumination, which is why we extract it directly as the side-output of N_{θ_1}. Here, we follow a concept from NeRF-w [34] that combines an embedding of the estimated illumination with the latent color vector produced by N_{θ_1}. As the dimensionality of the SGs can be large, we add a compaction network (N_{θ_2}), which encodes the 24×7 dimensional SGs to 16 dimensions. The compacted SG's embedding is then appended to the output of the last layer of N_{θ_1} and jointly passed to the final estimation network N_{θ_3} that outputs color values. Without the illumination dependent color prediction, several floaters would appear in the volume estimate, introducing wrong semi-transparent geometry to paint in highlights for individual views (see Fig. 3b). By introducing the illumination-dependent branch, the resulting 3D volume is sparser and more consistent.

Decomposition network. After a ray has sampled the *sampling network*, additional m samples are placed based on the density σ. This is visualized in Fig. 2b as the additional green points on the ray. The decomposition network is trained with the same loss as the *sampling network*. However, we introduce an explicit decomposition step and a rendering step in-between. Our decomposition step estimates view and illumination independent BRDF parameters $\boldsymbol{b}$ and a surface normal $\boldsymbol{n}$ at each point. The popular Cook-Torrance analytical BRDF model [14] is used for rendering. Here, we choose the Disney BRDF Basecolor-Metallic parametrization [11] instead of independently predicting the diffuse and specular color, as it enforces physical correctness. The illumination $\boldsymbol{\Gamma}^j$, in the form of spherical Gaussians (SG), is also jointly optimized. After rendering the decomposed parameters, the final output is a view and illumination dependent color $\boldsymbol{c}^j_{\omega_o r}$.

By keeping the rendering differentiable, the loss from the input color $\hat{\boldsymbol{c}}_r^j$ can be backpropagated to the BRDF $\boldsymbol{b}$, the normal $\boldsymbol{n}$, and the illumination $\boldsymbol{\Gamma}^j$. Our rendering step approximates the general rendering equation $L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega_i}, \boldsymbol{\omega_o}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i})(\boldsymbol{\omega_i} \cdot \boldsymbol{n}) d\boldsymbol{\omega_i}$ using a sum of 24 SG evaluations. The $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_o}$ defines the incoming and outgoing ray direction, respectively. The reflectance due to diffuse and specular lobes is separately evaluated by functions ρ_d and ρ_s, respectively [51]. Overall, our image formation is defined as: $L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) \approx \sum_{m=1}^{24} \rho_d(\boldsymbol{\omega}_o, \boldsymbol{\Gamma}_m, \boldsymbol{n}, \boldsymbol{b}) + \rho_s(\boldsymbol{\omega}_o, \boldsymbol{\Gamma}_m, \boldsymbol{n}, \boldsymbol{b})$. Our differentiable rendering implementation follows Boss *et al.* [10].

The overall network architecture is shown in Fig. 2b. Especially in the early stages of the estimation, joint optimization of BRDF and shape proved difficult. Therefore, we estimate the density σ and, in the beginning, a view-independent color $\boldsymbol{d}$ for each point in N_{ϕ_1}. The direct color prediction $\boldsymbol{d}$ is compared with the input image, and the loss is faded out over time when the rough shape is established.

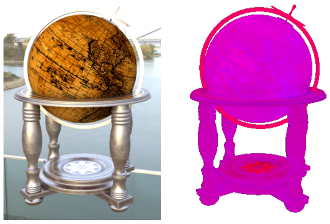
To compute the shading, the surface normal is required. One approach could be to simply learn the normal as another output [6]. However, this typically leads to inconsistent normals that do not necessarily fit the object's shape (Fig. 3c). Specific reflections can be created by shifting the normal instead of altering the BRDF. Coupling the surface normal to the actual shape can resolve some of this ambiguity [10]. In coordinate-based volume representations like NeRF [36], we can establish this link by defining the normal as the normalized negative gradient of the density field: $\boldsymbol{n} = -\frac{\nabla_{\boldsymbol{x}}\sigma}{\|\nabla_{\boldsymbol{x}}\sigma\|}$. While the density field defines the surface implicitly, the density in the 3D volume changes drastically at the boundary between non-opaque air to the opaque object. Thus, the gradient at a surface will be perpendicular to the implicitly represented surface. This is a similar to the normal reconstruction from SDFs of Yariv *et al.* [56].

By calculating the gradient inside the optimization and allowing the photometric loss from the differentiable rendering to optimize the normal, we optimize the σ parameter in the second order. Therefore, the neighborhood of surrounding points in the volume is smoothed and made more coherent with the photometric observations. As a more densely defined implicit volume allows for a smoother normal, we additionally jitter the ray samples during training. Each ray is now cast in a subpixel direction, and the target color is obtained by bilinear interpolation.

For the BRDF estimation, we use the property that often real-world objects consist of a few highly similar BRDFs which might be spatially separated. To account for this we introduce an additional network N_{ϕ_2} which receives the latent vector output of N_{ϕ_1}. This autoencoder creates a severe bottleneck, a two-dimensional latent space, which encodes all possible BRDFs in this scene. As the embedding space enforces a compression, similar BRDFs will share the same embedding. This step couples the BRDF estimation of multiple surface points, increasing the robustness. The assignment to various BRDFs is visualized Fig. 3a, which can be utilized for material-based segmentation.

The approach will converge to a globally consistent state, as the underlying shape and BRDF is assumed to be the same for all input images. The SGs are estimated for each input image, but we can force them to be the same or a rotated version of a single SG in case of static illumination.

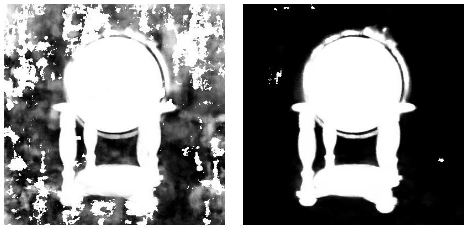
Dynamic range, tonemapping and whitebalancing. As



Input

Material Map

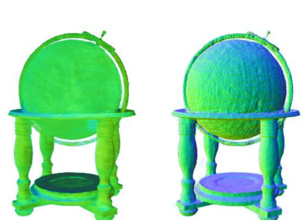
Figure 3a: **Compressed BRDF Space.** Instead of directly estimating the BRDF, we learn a 2D embedding per scene which clusters similar materials. As several points jointly estimate BRDFs, this stabilizes the decomposition and improves quality. Notice how similar materials are identified across the surface in the resulting material map.



Direct color prediction

Considering illumination

Figure 3b: **SGs dependent Sampling Network.** N_{θ_1} can to some extent model view-dependence by composition along the ray. This, however, is too weak to deal with varying illumination. Apparent highlights introduce spurious geometry that mimics the effect for individual views. We can obtain better shape by estimating the illumination dependent radiance with Spherical Gaussians (SG).



Directly Predicted

$\nabla\sigma$-based

Figure 3c: **Surface Normal Estimation.** Instead of directly predicting the normal as another output of the N_{θ_1}, the normal in our approach is calculated from the gradient of the density $\nabla\sigma$. Photometric information thus influences both $\boldsymbol{n}$ and σ during training.

most online image collections consist of Low Dynamic Range (LDR) images with at least an sRGB curve and white balancing applied, we have to ensure that our rendering setup's linear output recreates these mapping steps before computing a loss. However, rendering can produce a large value range depending on the incident light and the object's specularity. Real-world cameras also face this problem and tackle it by changes in aperture, shutter speed, and ISO. Based on the meta-data information encoded in JPEG files, we can reconstruct the input image's exposure value and apply this to our re-rendering. NeRD is then forced to always work with physically plausible ranges. For synthetic examples, we calculate these exposure values based on Saturation Based Sensitivity auto exposure calculation [1] and also apply an sRGB curve.

Cameras also apply a white balancing based on the illumination, or it is set by hand afterward. This can reduce some ambiguity between illumination and material color and, in particular, fixes the overall intensity of the illumination. For synthetic data, we evaluate a small spot of material with 80% gray value in the environment. We assume a perfect white balancing and exposure on real-world data, at least for one of the input images. The RGB color ($\boldsymbol{w}$) of the white point is stored. After each training step a single-pixel with a rough 80% gray material is rendered in the estimated illumination and a factor $\boldsymbol{f} = \frac{\boldsymbol{w}}{\boldsymbol{b}}$ is calculated. This factor is then applied to the corresponding SG. As the training will adopt the BRDF to the normalized SG, a single white-balanced input can implicitly update and correct all other views. In practice, the calculated factor $\boldsymbol{f}$ could change the SGs abruptly in one step causing unstable training. Therefore, we clip the range of $\boldsymbol{f}$ to $[0.99; 1.01]$ to spread the update over multiple training iterations.

Mesh extraction. The ability to extract a consistent textured mesh from NeRD after training is one of the key advantages of the decomposition approach and enables real-time rendering and relighting. This is not possible with NeRF-based approaches where the view-dependent appearance is directly baked into the volume. The basic process generates a point cloud, computes a mesh, including a texture atlas, and then fills the texture atlas with BRDF parameters. More details are given in the supplementary material.

Training and losses. The estimation is driven by a Mean Squared Error (MSE) loss between the input image and the results of evaluating randomly generated rays. For the *sampling network* this loss is applied to the RGB prediction and for the *decomposition network* to the re-rendered result $\boldsymbol{c}^j$ and the direct color prediction $\boldsymbol{d}$. The loss for the color prediction based on $\boldsymbol{d}$ is exponentially faded out. Additionally, we leverage the foreground/background mask as a supervision signal, where all values along the ray in background regions are forced to 0. This loss is exponentially faded in throughout the training to reduce optimization instabilities. By gradually increasing this loss, the network is forced to provide a more accurate silhouette, which prevents the smearing of information at the end of the training. The networks are trained for 300K steps with the Adam optimizer [25] with a learning rate of $5\mathrm{e}{-4}$. On 4 NVIDIA 2080 Ti, the training takes about 1.5 days. The final mesh extraction takes approximately 90 minutes.

4. Results

The proposed method recovers shape, appearance, and illumination for relighting in unconstrained settings. Our reconstruction and relighting performance on synthetic sets is measured against ground truth images and known BRDF parameters. For real-world examples, we present novel, relit views and compare the renderings with validation images excluded from training. If the environment map for the validation image is known, we directly use this for relighting.

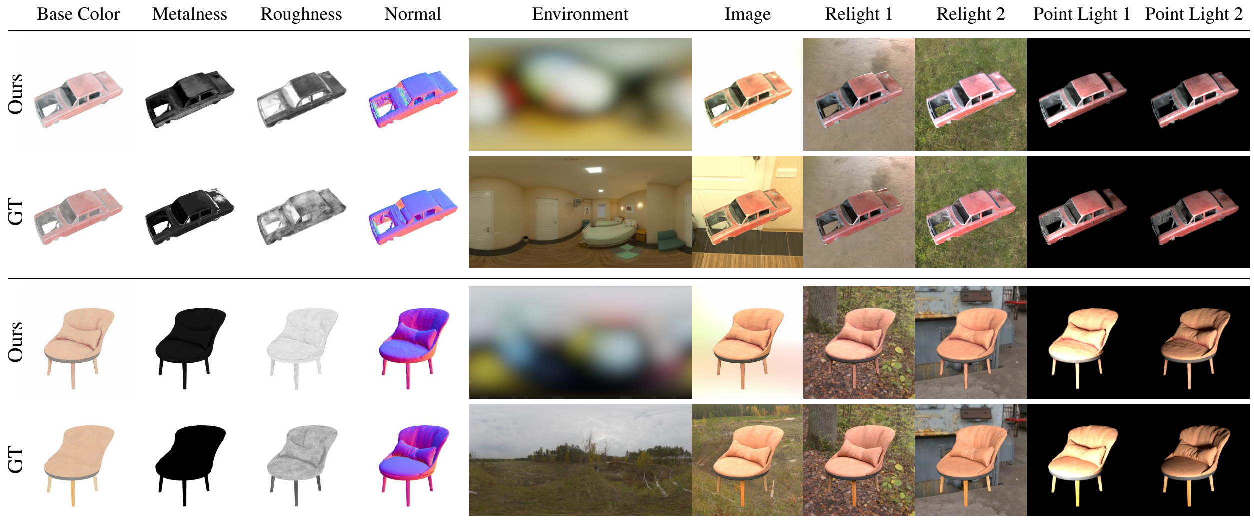


Figure 4: **Decomposition on Synthetic Examples.** Two scenes are highlighted to show the decomposition performance of our method. Notice the accurate performance in relighting with unseen illuminations.

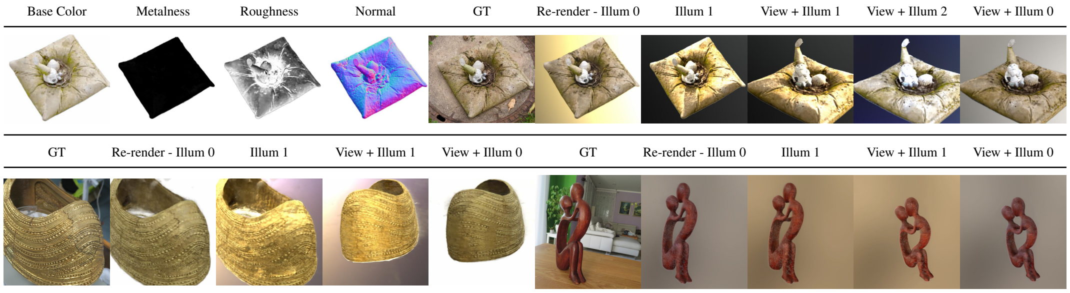


Figure 5: **Real World BRDF Decomposition and Relighting.** The decomposition produces plausible BRDFs, and re-rendered images are close to the ground truth input images. Note that the estimated parameters are hardly affected by the shadows visible in the input of the gnome scene. When relit with the estimated SG of a validation view, the appearance is well reproduced. Even in a different perspective or under completely novel, artificial illumination the recovered BRDF parameters result in convincing images.

Otherwise, we recover the unseen illumination by optimizing the SGs through the frozen network in 1000 steps using stochastic gradient descent with a learning rate of 0.1. One-to-one comparisons with previous methods are challenging, as most methods use different capturing setups. We can, however, compare to the outcome of NeRF when trained on a similar scene. NeRF cannot relight the object under novel illumination, and even NeRF-w and our simplified NeRF-A baseline can only interpolate between seen illuminations.

We also perform an ablation study to study the influence of our novel training techniques. We refer to the supplementary material for additional results and the extracted textured meshes for different scenes.

Datasets. We use three synthetic scenes to showcase the quality of the estimated BRDF parameters. We use three textured models (Globe [50], Wreck [12], Chair [13]) and render each model with a varying environment illumination per image. For a fixed illumination synthetic dataset, we use the Lego, Chair and Ship scenes from NeRF [36].

We also evaluate using two real-world scenes from the British Museum's photogrammetry dataset: an Ethiopian Head [39] and a Gold Cape [40]. These scenes feature an object in a fixed environment with either a rotating object or a camera. Additionally, we captured our own scenes under varying illumination at various times of day (Gnome, MotherChild).

BRDF decomposition results. Fig. 4 shows exemplary views and decomposition results of the synthetic Car Wreck and Chair scenes. In all cases, we observe the estimated re-renderings to be very similar to GT. The estimated BRDF parameters may not match perfectly in some places compared to the GT, but given the purely passive unknown illu-

Method [PSNR↑]	Diffuse	Specular	Roughness
Li *et al.*	1.06	—	17.18
Li *et al.* + NeRF	1.15	—	**17.28**
Ours	**18.24**	**25.70**	15.00

Table 1: **BRDF estimation.** Comparison with a recent state-of-the-art method in BRDF decomposition under environment illumination [32]. Li *et al.*: directly on test images, Li *et al.* + NeRF: NeRF trained on BRDFs from [32].

		Fixed Illumination		Varying Illumination	
	Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Syn.	NeRF	**34.24**	**0.97**	21.05	0.89
	NeRF-A	32.44	**0.97**	**28.53**	0.92
	Ours	30.07	0.95	27.96	**0.95**
Real	NeRF	23.34	0.85	20.11	0.87
	NeRF-A	22.87	0.83	**26.36**	0.94
	Ours	**23.86**	**0.88**	25.81	**0.95**

Table 2: **Novel view synthesis.** Comparison with NeRF and NeRF-A on novel view synthesis (with relighting in varying illumination). Notice NeRF and NeRF-A is not capable of relighing in any unseen illuminations, nor is an extraction of a textured mesh from the network easily possible.

mination setup, they still reproduce the GT images. Causes for deviations are the inherent ambiguity of the decomposition problem as well as the differences in shading based on SG *vs*. the high-resolution GT environment map.

Several sub-tasks of the unconstrained shape and BRDF decomposition problem have been addressed by earlier works. Unfortunately, trying to recover parameters separately or sequentially, *e.g*. geometry, BRDF, or illumination, often fails in challenging scenes. We show that COLMAP fails to reconstruct a plausible geometry for some of our data sets in the supplementary material . If the following stages rely on accurate geometry, the pipeline cannot recover meaningful material properties from the inaccurate shape. We also tried recovering the BRDF parameters (diffuse and roughness) for each image separately using the work of Li *et al.* [32] followed by NeRF to handle the view interpolation. To run on view independent BRDF parameters, we adapted NeRF accordingly. However, NeRF fails to create a coherent geometry, as each image results in drastically different BRDF parameter maps.

We, therefore, conclude that joint optimization of shape and SVBRDF is essential for this extremely ambiguous problem. Quantitative comparisons with Li *et al.* are shown in Table 1. These are average PSNR results over our synthetic datasets (Globe, Wreck, and Chair). We decompose our basecolor into *diffuse* and specular to enable comparison with Li *et al.*, which uses a *diffuse* and roughness parameterization. It is worth noting that Li *et al.* here is a weak baseline, but the closest available, as their method expects a flash light in conjunction to the environment illumination. However, as most scenes are captured with an outside environment illumination, the flash will be barely noticeable due to the strong sun light.

Relighting and novel view synthesis. In Fig. 5, novel views and plausible relighting in unseen environments are shown for our real-world data sets. The relighted images are visually close to the held-out validation images. Furthermore, a novel view can be relighted with the lighting from a different view. Note that some fine details are missing in the reconstructions of the Gold Cape, which is caused by small inaccuracies in the camera registration. Also the MotherChild model is missing some highlights especially at grazing angles, which can be attributed to the limitations of the SG based rendering model.

While no ground truth BRDF exists, the estimated parameters for the Gnome (Fig. 5) seems plausible. The material is correctly classified as non-metallic (black metalness map), has a higher roughness, and the normal also aligns well with the shape. In the central valley, where dirt is collected, the BRDF parameters increase in roughness compared to the clean, smooth concrete pillow surface. The color is also captured well, and in re-rendering, the similarity to the ground-truth is evident.

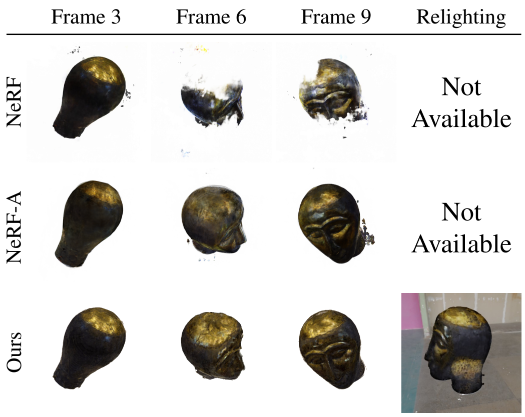


Figure 6: **Novel View Comparison with NeRF and NeRF-A on real-world Ethiopian Head.** Notice the improved consistency in our method. NeRF introduces highlights as floaters in the radiance volume that inconsistently occlude the scene geometry in other views. Additionally, we showcase the quality in relighting the head with our method.

Another evaluation focuses on the use of our method purely for novel view synthesis, with implicit relighting. In this setting, our method can be compared with NeRF [36] and also an extension to NeRF, called NeRF-A, which inspired from [34]. NeRF-A models the appearance change per image in a 48 dimensional latent vector. It is worth noting that NeRF-A is a strong baseline as the task is simpler compared to NeRD and it is only capable of relighting within known scenes. On fixed illumination scenes, NeRF-A is not capable of relighting. Table 2 shows the quantitative results over multiple datasets, real-world (Real) and synthetic (Syn.), on the test views wherein the "Fixed Illumination" case only novel view synthesis is performed, and in the "Varying Illumination" case, novel view synthesis and relighting. Here, "Varying Illumination" also refers to case where the object is rotating w.r.t the camera and therefore the relative illumination is varying. The corresponding datasets for the fixed and synthetic case are from NeRF (Ship, Lego, Chair), and for varying, we use ours (Globe, Wreck, Chair). For the real-world comparison, Cape provide fixed illumination, and the Gnome, Head and Mother-Child scenes are recorded in varying environments. PSNR and SSIM results show that NeRF performs quite poorly in varying illumination cases. NeRF-A on the other hand is a strong baseline in the varying illumination case which we mostly match or surpass while solving a more challenging problem which allows for more flexible relighting use cases.

Fig. 6 shows the novel view synthesis results of NeRF, NeRF-A and NeRD (Ours) on the Ethiopian Head real-world scene. The object rotates in front of the camera. We, therefore, compose the Head on a white background in Fig. 6 as NeRF cannot handle a static background with a fixed camera. During training, both models recreate the input quite closely. However, in the test views, NeRF added spurious geometry to mimic highlights for specific camera locations, which are not seen by other cameras in the training set. NeRF-A can expess the relative illumination change in the appearance embedding and can improve the reconstruction quality compared to NeRF. However, as only a single illumination type is seen NeRF-A is still not capable of relighting under arbitrary illumination. Due to our physically motivated setup with the explicit decomposition of shape, reflectance, and illumination, these issues are almost completely removed. Our method creates convincing object shapes and reflection properties, which, in addition, allow for relighting in novel settings.

Overall, it is evident that NeRF will not work with varying illuminations, clearly demonstrating the advantage of our more flexible decomposition. It is also worth noting that even if an appearance embedding as in NeRF-w [34] our our simplified NeRF-A baseline is used, the method can only interpolate between seen illuminations. Our model is capable of relighting even if the scene was only captured in a single fixed illumination.

Ablation study. In Table 3, we ablate the gradient-based normal estimation, the BRDF interpolation in a compressed space, and incorporating the white balancing in the optimization. We perform this study on the Globe scene as it contains reflective, metallic, and diffuse materials and fine geometry. One of the largest improvements stems from the addition of gradient-based normals. The coupling of shape and normals improves the BRDF and illumination separation. Normals cannot be rotated freely to mimic specific reflections. The compressed BRDF space also improves the result, especially in the metalness parameter estimation. This indicates that the joint optimization of the encoder/decoder network N_{ϕ_2} effectively optimizes similar materials across different surface samples. The white balancing fixes the absolute intensity and color of the SGs, which indirectly forces the BRDF parameters into the correct range.

Method	Base Color	Metalness	Roughness	Normal	Re-Render
w/o Grad. Normal	0.1264	0.1203	0.3192	0.1664	0.0893
w/o Com. BRDF	0.1828	0.2496	0.2827	0.0089	0.0759
w/o WB	0.1059	0.0870	0.2754	0.0087	0.0655
Full Model	**0.0796**	**0.0784**	**0.2724**	**0.0084**	**0.0592**

Table 3: **Ablation Study.** The MSE loss on 10 test views with ablation of gradient (grad.) normals, compressed (Com.) BRDF and white balancing (WB) on the globe dataset.

5. Conclusion

In this work, we tackle an extremely challenging problem of decomposing shape, illumination, and reflectance by augmenting coordinate-based radiance fields with explicit representations for the BRDF and the illumination. This decomposition renders our approach significantly more robust than simple appearance-based representations, or other multi-view stereo approaches w.r.t. changes in the illumination, cast shadows, or glossy reflections. Additionally, we propose a method to link the surface normal to the object's actual shape during optimization. This link allows a photometric loss to alter the shape by backpropagation through differentiable rendering. Our method enables realistic real-time rendering *and* relighting under arbitrary unseen illuminations via explicit mesh extraction from the neural volume.

While the results from the method are convincing, there exists several limitations. Currently, no explicit shadowing is modeled while the object is optimized. Especially in scenes with a static environment illumination and deep crevices, a shadow will be baked into the diffuse albedo. Additionally, the chosen SGs environment model helps in a stable and fast shading evaluation but is often limiting when high-frequency light effects are present in a scene. A different, maybe implicit environment representation might produce better results, but it would need to support efficient BRDF evaluation.

Acknowledgement This work was funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) - Projektnummer 276693517 - SFB 1233 and by EXC number 2064/1 – Project number 390727645.

References

[1] Technical Committee ISO/TC 42. Photography — Digital still cameras — Determination of exposure index, ISO speed ratings, standard output sensitivity, and recommended exposure index. Standard, International Organization for Standardization, 2019. 5

[2] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. In *ACM Transactions on Graphics (ToG)*, 2018. 2

[3] Rachel Albert, Dorian Yao Chan, Dan B. Goldman, and James F. O'Brian. Approximate svBRDF estimation from mobile phone video. In *Eurographics Symposium on Rendering*, 2018. 2

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

[5] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015. 1

[6] 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 e-prints*, 2020. 1, 2, 4

[7] 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 (ECCV)*, 2020. 1, 2

[8] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[9] Mark Boss, Fabian Groh, Sebastian Herholz, and Hendrik P. A. Lensch. Deep Dual Loss BRDF Parameter Estimation. In *Workshop on Material Appearance Modeling*, 2018. 1, 2

[10] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot spatially-varying brdf and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4

[11] Brent Burley. Physically based shading at disney. In *ACM Transactions on Graphics (SIGGRAPH)*, 2012. 4

[12] cgtrader. Carwreck. `https://www.cgtrader.com/free-3d-models/vehicle/other/car-wreck-pbr-game-asset`. 6

[13] cgtrader. Chair. `https://www.cgtrader.com/free-3d-models/furniture/chair/freifrau-easy-chair-pbr`. 6

[14] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1982. 4

[15] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. In *ACM Transactions on Graphics (ToG)*, 2018. 2

[16] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. In *Eurographics Symposium on Rendering*, 2019. 2

[17] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. In *Eurographics Symposium on Rendering*, 2020. 2

[18] Yue Dong, Guojun Chen, Pieter Peers, Jianwen Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2014. 2

[19] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2019. 1, 2

[20] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (ToG)*, 2017. 1

[21] Dan B. Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009. 1

[22] Tom Haber, Christian Fuchs, Phillipe Bekaer, Hans-Peter Seidel, Michael Goesele, and Hendrik P. A. Lensch. Relighting objects from image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[23] James T. Kajiya. The rendering equation. In *ACM Transactions on Graphics (SIGGRAPH)*, 1986. 1

[24] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ArXiv e-prints*, 2014. 5

[26] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient brdf importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 2004. 1, 2

[27] Hendrik Lensch, Jan Kautz, Michael Gosele, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Conference on Rendering*, 2001. 1, 2

[28] Hendrik P.A. Lensch, Jochen Lang, M. Sa Asla, and Hans-Peter Seidel. Planned sampling of spatially varying brdfs. In *Computer Graphics Forum*, 2003. 1, 2

[29] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. In *ACM Transactions on Graphics (ToG)*, 2017. 2

[30] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[31] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[32] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018. 7

[33] Lingjie Liu, iatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[34] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *ArXiv e-prints*, 2020. 2, 4, 7, 8

[35] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[36] Ben Mildenhall, Pratul Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 4, 6, 7

[37] Giljoo Nam, Diego Gutierrez, and Min H. Kim. Practical SVBRDF acquisition of 3d objects with unstructured flash photography. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018. 1, 2

[38] Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 1965. 1

[39] Daniel Pett. Ethiopian Head, 2016. 6

[40] Daniel Pett. BritishMuseumDH/moldGoldCape: First release of the Cape in 3D, Mar. 2017. 6

[41] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *ACM Transactions on Graphics (SIGGRAPH)*, 2001. 1

[42] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[43] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[44] 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. 2

[45] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[46] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[47] Vincent Sitzmann, Justus Thies, Felix Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[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 *ArXiv e-prints*, 2020. 2

[49] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[50] TurboSquid. Standing Globe. https://www.turbosquid.com/3d-models/3d-standing-globe-1421971. 6

[51] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2009. 4

[52] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2016. 2

[53] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[54] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 2019. 1

[55] Zexiang Xu et al. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 2018. 1

[56] 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 (NeurIPS)*, 2020. 2, 4

[57] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Single image surface appearance modeling with self-augmented cnns and inexact supervision. *Computer Graphics Forum*, 2018. 2

[58] Jianzhao Zhang, Guojun Chen, Yue Dong, Jian Shi, Bob Zhang, and Enhua Wu. Deep inverse rendering for practical object appearance scan with uncalibrated illumination. In *Advances in Computer Graphics*, 2020. 2

[59] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[60] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *ArXiv e-prints*, 2020. 2