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

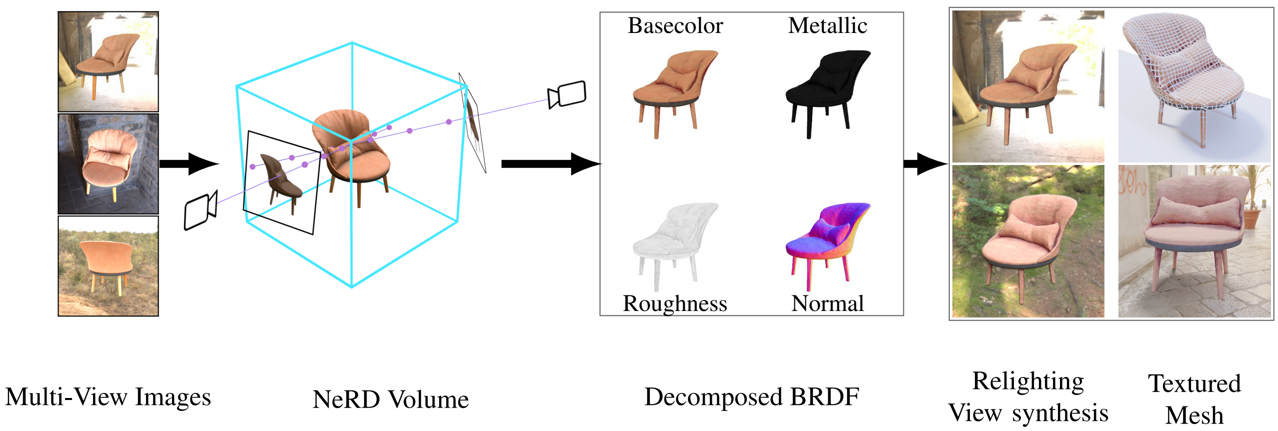


Figure 1: **Neural Reflectance Decomposition.** Multiple views of an object under varying or fixed illumination are encoded into the NeRD volume. During the encoding process, information provided by all samples is decomposed into geometry, spatially-varying BRDF parameters and a rough approximation of the incident illumination in a globally consistent way. This decomposition can be easily extracted and re-rendered under a novel illumination condition in real-time.

Abstract

Decomposing a scene into its shape, reflectance, and illumination is a challenging but essential 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, these techniques only enable view synthesis and not relighting. Additionally, evaluating these radiance fields is resource and time-intensive. By decomposing a scene into explicit representations, any rendering framework can be leveraged to generate novel views under any illumination in real-time. NeRD is a method that achieves this decomposition by introducing physically-based rendering to neural radiance fields. Even challenging non-Lambertian reflectances, complex geometry, and unknown illumination can be decomposed to high-quality models. The datasets and code is 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 [13, 67], relighting [6, 13, 25, 26, 38, 68], object insertion into environments [10, 24, 38] *etc*. This problem is often referred to as *inverse rendering* [30, 53], 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) [50].

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 [28]. 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, 12, 33–35]. More recent approaches for SVBRDF estimation employ more practical capture setups [9–11, 13, 23, 47], 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) [44, 46, 72] 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. Color and density estimates along the ray are obtained by stepping through the volume, querying the network at each sample location. Though these approaches are capable of reproducing semitransparent and view-dependent appearance effects, the radiance of a point in a direction is "baked in" to these networks, making them unusable for relighting. NeRF also assumes a static illumination across its input images, which limits its practicality in uncontrolled settings where illumination may vary across images (though "NeRF-in-the-Wild" largely ameliorates this particular issue [44]). Even if such techniques could be extended to relighting, the rendering speed of these radiance field approaches limits their practicality — the time required by NeRF to generate a single view is about 30 seconds [46] on a modern GPU system.

In this work, we present a shape and SVBRDF estimation technique that allows for a more flexible capture setting while enabling relighting under novel illuminations and supports fast rendering. Our key technique is an explicit decomposition model for shape, reflectance, and illumination (and a corresponding differentiable rendering engine) within a NeRF-like coordinate-based neural representation framework [46]. While our volumetric geometry representation is similar to previous approaches, we explicitly represent the appearance at each point in the volume as a function of spatially-varying BRDF parameters and a spherical Gaussian mixture model for the illumination. That is, instead of directly solving for the outgoing radiance at each point in the volume, we solve for SVBRDF parameters at each 3D point and a global scene illumination for each input image (see Figure 1). With this representation, we can explicitly evaluate the outgoing radiance at each point as analytical integrals of the product of BRDF and illumination using a differentiable formulation. This integrates a physically plausible image formation process into the scene representation networks, thereby constraining the solution space. 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. In addition, 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 (*e.g*. COLMAP [55, 56]), 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 observation for varying illuminations or globally for static illumination.

Another key technique that we use in NeRD is the direct estimation of surface normals from our coordinate-based density representation. That is, instead of optimizing surface normals independently of the recovered geometry [9], we derive normal directions using the gradients of the density field. By allowing gradients to backpropagate from the normal to the density field, changes to the surface normal due to the photometric loss can alter the density field and, therefore, the object's underlying shape. 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 [42, 44–46, 59, 60, 66, 72]. These methods surpassed previous state-of-the-art methods 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 [46]. 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.

Intrinsic images. Intrinsic images describes the task of separating a scene into its underlying physical constituents, such as reflectance and shading [5–7, 43, 62]. Though effective single-image techniques exist [6, 62], the underconstrained nature of this problem means that multi-image approaches are often more effective [27, 32, 69]. In single-image estimation, the inherent ambiguity between shape, shading, and reflectance is often tackled with deep learning approaches from labeled data [36, 48, 58], unlabeled [40] and partially labeled data [8, 39, 49, 74]. However, the use cases for the simplistic rendering model are limited compared to the full rendering equation.

BRDF estimation. Estimating the physical reflectance properties from images, *e.g*. BRDF parameters, is an active research topic. Though highly accurate BRDF measurements can be achieved under laboratory conditions with known view and light positions [4, 12, 33–35], 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, 19, 41, 54], few-shot [2] or multi-shot [3, 12, 20, 21, 23] estimation. This casual setup can be extended to estimating the BRDF and shape of objects [10, 11, 13, 47, 71] or scenes [57]. Recently, Bi *et al*. [9] 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 [37, 70] 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*. [22] by explicitly optimizing for illumination and reflectance from temporal appearance traces of rotating objects. Later, geometry reconstruction was added to the process [65]. However, these methods require images of rotating objects, which can be infeasible when capturing large objects. Our method supports more flexible capture settings.

3. Method

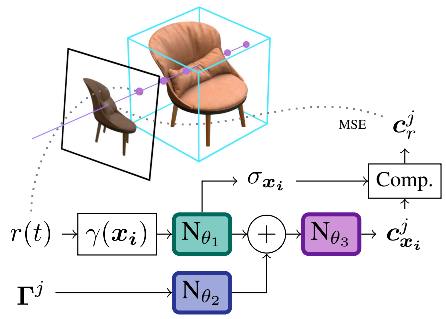
Our method jointly optimizes a model for shape, BRDF, and illumination by minimizing the photometric error to multi-view input images that can be captured under different illuminations.

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 $\boldsymbol{\Gamma} \in \mathbb{R}^{24\times 7}$.

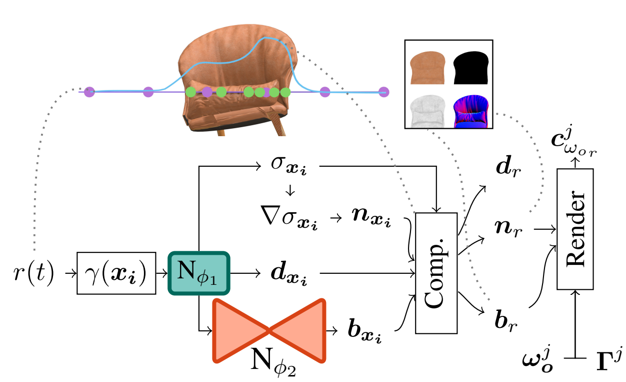
Preliminaries. We follow the general architecture of NeRF [46]. NeRF creates a neural volume for novel view synthesis using two Multi-Layer-Perceptrons (MLP). NeRF model encodes view-dependent color and object density information at each point in 3D space using MLPs. NeRF consists of two MLPs in which 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 $\hat{\boldsymbol{c}}^j_r$. Marching along the ray through the volume at each sample coordinate $\boldsymbol{x} = (x, y, z)$, the networks $\mathrm{N}_{(.)}$ 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*. [61] and Mildenhall *et al*. [46], 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 $T(t) = \exp\left(-\int_{t_n}^{t}\sigma(s)\,ds\right)$ [46], 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 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: $\frac{1}{s}\sum^s(\hat{\boldsymbol{c}}^j_r - \boldsymbol{c}^j_r)^2$. 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



(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 $\mathbf{\Gamma}^j$ from N_{θ_2} with the latent color output of N_{θ_1} to generate the illumination-dependent color in in N_{θ_3}.



(b) **Decomposition Network.** With the sampling pattern generated from the coarse network the actual decomposition is performed. 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 [46], and $\mathbf{\Gamma}^j$ denotes the Spherical Gaussian parameters per image j. $\boldsymbol{c}$ is the output color for 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.".

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 [44] 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 considerably 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 estimate view and illumination independent BRDF parameters $\boldsymbol{b}$ and a surface normal $\boldsymbol{n}$ at each point. The popular Cook-Torrance analytical BRDF model [18] is used for rendering. Here, we choose the Disney BRDF Basecolor-Metallic parametrization [14] instead of independently predicting the diffuse and specular color, as it enforces physical correctness. The illumination $\mathbf{\Gamma}^j$, 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 $\boldsymbol{c}^j_{\omega_{or}}$.

By keeping the rendering differentiable the loss from the input color $\hat{\boldsymbol{c}}^j_r$ can be propagated to the BRDF $\boldsymbol{b}$, the normal $\boldsymbol{n}$, and the illumination $\mathbf{\Gamma}^j$. 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 at a surface, respectively. The reflected radiance due to diffuse and specular lobes is separately evaluated by functions ρ_d and ρ_s, respectively [64]. Overall, our image formation is defined as: $L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) \approx \sum_{m=1}^{24} \rho_d(\boldsymbol{\omega}_o, \mathbf{\Gamma}_m, \boldsymbol{n}, \boldsymbol{b}) + \rho_s(\boldsymbol{\omega}_o, \mathbf{\Gamma}_m, \boldsymbol{n}, \boldsymbol{b})$.

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 [9]. 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 [13]. In coordinate-based volume representations like NeRF [46] 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 only 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 always be perpendicular to the implicitly represented surface.

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.

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

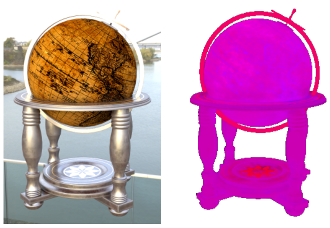
Environment representation and rendering. A Spherical Gaussian (SG) is defined as $G(\boldsymbol{\omega}; \boldsymbol{\mu}, \lambda, \boldsymbol{\alpha}) = \boldsymbol{\alpha} \exp^{\lambda(\boldsymbol{\mu}\boldsymbol{\omega}-1)}$, where $\boldsymbol{\mu}$ defines the unit length axis, $\boldsymbol{\alpha}$ the amplitude as RGB color values in our case, λ the width of the lobe, where larger values increase the sharpness and $\boldsymbol{\omega}$ the direction to evaluate. SGs are useful for representing low-frequency approximations of the environment. The advantage of SGs is that they are simple to evaluate and have the property that the inner product of two SGs is the integral of both. This is especially useful because the Cook-Torrance BRDF [18] can be approximated as two SGs [64]. Our differentiable rendering step follows the rendering implementation of Boss *et al*. [13].

Dynamic range, tonemapping and whitebalancing. As 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 or a picture of a gray card, 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 ($\boldsymbol{b}$) with a rough 80% gray material is rendered in the estimated environment 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.

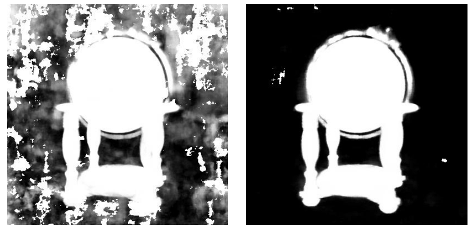
Mesh extraction. The ability to extract a consistent textured mesh from NeRD after training is one of the key advantages of the decomposition approach and enables real-time rendering and relighting. This is not possible with NeRF-based approaches where the view-dependent appearance is directly baked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



Input Material Map

Figure 3a: **Input Image and Compressed BRDF Space.** Instead of directly estimating the BRDF, we learn a 2D embedding per scene which clusters similar materials. As several points now jointly estimate BRDFs, our decomposition is more stable and the quality is improved drastically. 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.** The output of N_{θ_1} is per view and illumination independent. Some view dependence can be achieved by composition along the ray. This, however, is too weak to deal with varying illumination. In this case, apparent highlights might introduce spurious geometry that mimics the effect for individual views. By passing the Spherical Gaussians (SG) to the network, the estimated radiance is now illumination dependent and the volumetric space is cleaner.

Directly Predicted $\nabla\sigma$-based

Figure 3c: **Surface Normal Estimation.** Instead of predicting the normal by training a direct output of the N_{θ_1} the normal in our approach is calculated from the gradient of the density σ. Photometric information thus influences both $\boldsymbol{n}$ and σ.

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 loss, wherein background regions, all values along the ray, are forced to 0. This loss is exponentially faded in throughout the training as an initial optimization leads to 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. In each batch, 1024 random rays of one view are selected. The networks are trained for 300.000 steps with the Adam optimizer [31] with a learning rate of $5\mathrm{e}{-4}$. On an NVIDIA 2080 Ti, the training takes about two days. The final mesh extraction can be done in approximately 90 minutes.

4. Results

The proposed method tackles the otherwise rarely investigated problem of recovering shape, appearance, and illumination for relighting in unconstrained settings. One to one comparison with previous method are challenging, as most methods use a different capturing setup. We can, however, compare to the outcome of NeRF when trained on a similar scene. NeRF is not capable of relighting the object under novel illumination, and even NeRF-w can only interpolate between seen illuminations.

To demonstrate the quality of our results, we present novel views on real-world data and compare the renderings with validation images. In addition, the quality of a re-rendering in a completely novel illumination is investigated. We can show the quality of novel views for synthetic scenes and calculate the error in the BRDF parameter predictions. An ablation study is performed to show the influence of several novel training techniques. We refer to the supplementary for additional results and the extracted textured meshes for the scenes.

Datasets. NeRD is a general method which can decompose any scene given sufficient images. Three synthetic scenes are selected to showcase the quality of the estimated BRDF parameters. Here, we selected three textured models (Globe [63], Wreck [15], Chair [16]) and render each model with a varying environment illumination per image.

Two real-world scenes from the photogrammetry dataset of The British Museum are used, an Ethiopian Head [51] and a Gold Cape [52]. These scenes feature an object in a fixed environment with either a rotating object or a camera. Additionally, we captured an own scene (Gnome) shown in Fig. 5 under varying illumination at various times of day.

Synthetic results. Fig. 7 shows exemplary views of the Car Wreck. In all cases, the overall error in the re-renderings is visually and quantitatively quite low. The optimization process estimated BRDFs, which yield convincing output renderings. These parameters may not match perfectly in some places compared to the ground truth, but given the purely passive unknown illumination setup, they still reproduce the input 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-res ground truth environment map. This can be seen in the roughness parameters, as here high-frequency shading emphasizes the specular highlights on the surface.

Real-world results. For the real-world evaluation, as for the samples of the Gnome in Fig. 4, no ground truth for the BRDF parameters is available. However, the input appearance is closely matched, which also reflects in the low

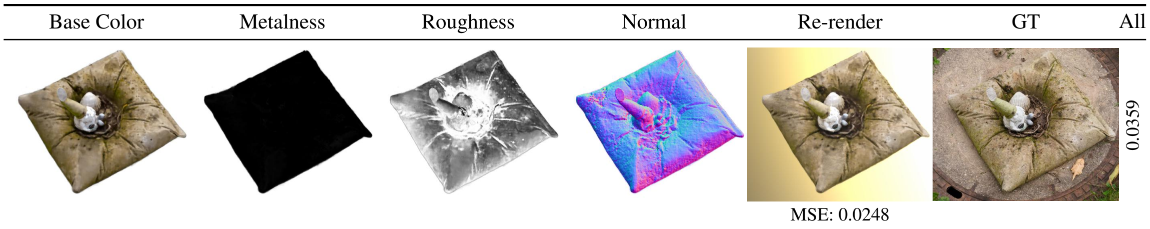


Figure 4: **Real World BRDF Decomposition.** The decomposition produces plausible BRDFs and re-rendered images are close to the ground truth input images. Note that the estimated parameters are hardly affected by the cast shadows visible in the input. The last column shows the overall MSE for all re-rendered images in the dataset.

Figure 5: **Gnome Dataset.** Examplary images of the Gnome Dataset, which was captured under varying illuminations and day times.

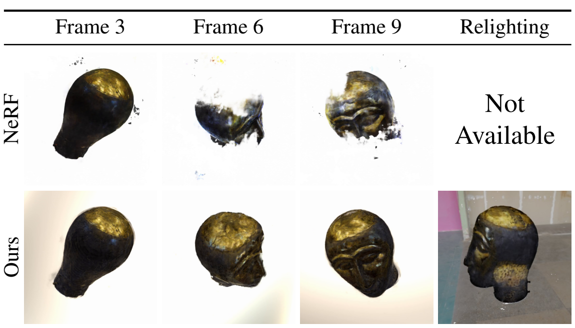


Figure 6: **Novel View Comparison with NeRF on real-world data.** Novel views of the Ethiopian Head are evaluated. Notice the improved consistency in our method. NeRF introduces highlights as floaters in the radiance volume that inconsistently occlude the scene geometry in other views. Additionally,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

MSE between the re-rendered and the input images. Overall, the optimization results in plausible BRDF parameters. The concrete pillow and clay gnome are estimated as non-metallic and with a larger roughness. In the central valley, where dirt is collected, the BRDF parameters increase in roughness compared to the clean, smooth concrete pillow surface. The shape is also captured perfectly, as seen in the normal map. Each individual fold of the pillow is visible and consistent.

Comparisons with NeRF. No direct method exists which handles decomposition on non-constrained capture settings. NeRF [46] only encodes the view-dependent final shading per point, *i.e.* the integral of the BRDF and the scene illumination. This implicit appearance model can encode even effects such as subsurface scattering or anisotropy. The model internally does not decompose this information but interpolates between the color information from various views. As the model does not explicitly extract the BRDF parameters, one can only judge the visual fidelity of the views generated from NeRF compared to our re-rendered predictions.

The Ethiopian Head real-world scene is selected for the comparison. The scene only features a single illumination while 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 optimization, the loss of the training images is quite comparable, showing only a slight improvement for our approach (MSE - NeRF: 0.0470 *vs*. ours: 0.0445). However, the differences in the quality of the models become visible in novel views. NeRF added many spurious geometry to mimic highlights for specific camera locations, which are not seen by other cameras in the training set. Due to our physically motivated set up 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.

Ablation study. The gradient-based normal estimation, the BRDF interpolation in a compressed space, and incorporating the white balancing in the optimization are novel techniques analyzed in an ablation study, evaluated in Table 1. The Globe scene is selected for this study, as it contains reflective and metallic materials, fine geometry, and diffuse surfaces. One of the largest improvements stems from the addition of gradient-based normals. The explicit coupling of shape and normals also improves the BRDF and illumination separation. Normals cannot be rotated freely to mimic specific reflections.

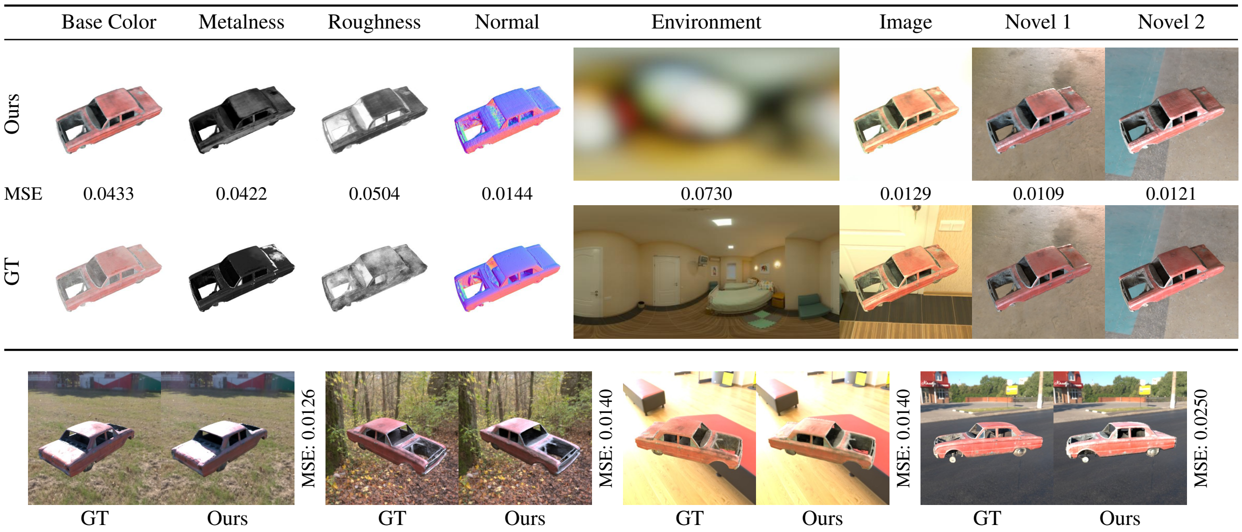


Figure 7: **Synthetic Examples.** The top portion shows an example of the decomposition and the bottom of the resulting re-rendered views. In the bottom, a visual and quantitative comparison is done on a test set of four unseen view positions and random seen illuminations. Notice the overall low error in re-rendering.

The compressed BRDF space also improves the result significantly, 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	Environment	Re-Render
Gradient Normal	0.1574	0.1203	0.3192	0.1664	0.1288	0.0893
compr. BRDF	0.1137	0.2496	0.2827	0.0089	0.0956	0.0759
Ours	0.1059	0.0870	0.2754	0.0087	0.0883	0.0655
Ours+WB	**0.0510**	**0.0784**	**0.2724**	**0.0084**	**0.0812**	**0.0592**

Table 1: **Ablation Study.** The MSE loss on ten test images for selected disabled novel additions.

Shadows and interreflections. So far, our method does not model indirect reflections nor shadows, and clearly, these effects cannot be correctly predicted. However, if the effects are not stationary and only appear in individual views, *e.g.*, due to changing illumination or glossy reflections, our approach still handles them well. Cast shadows in the Gnome Scene input (Fig. 4) or the glossy reflection of the leg of the Globe (Fig. 3a) are correctly ignored and not baked into the SVRDF as this would be inconsistent with many other input images. On the other side, if the self-shadowing is present in almost all images, *e.g.*, the darkening in the folds, the estimated BRDF parameters will be affected, corresponding to baked in ambient occlusion.

5. Conclusion

The proposed method solves 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 illumination after extracting an explicit surface.

While the results from the method are convincing, some limitations remain for future work. 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 albedo. Additionally, the chosen SGs environment model helps in a stable and fast shading evaluation but proofed limiting with high-frequency light effects 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 partly funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) - Projektnummer 276693517 - SFB 1233.

Supplementary

Implementation details. The main network N_{θ_1}/N_{ϕ_1} uses 8 MLP layers with a feature dimension of 256 and ReLU activation. The input coordinate $\boldsymbol{x}$ is transformed by the Fourier output $\gamma(\boldsymbol{x})$ with 10 bands to 63 features. For the *sampling network*, the output from the main network is then transformed to the density σ with a single MLP layer without any activation. The flattened 192 SGs parameters Γ^j are compacted to 16 features using a fully connected layer (N_{θ_2}) without any activation. As the value range can be large in the real-world illuminations, the amplitudes α are normalized to $[0, 1]$. The main network output is then concatenated with the SGs embeddings and passed to the final prediction network. Here, an MLP with ReLU activation is first reducing the joined input to 128 features. The final color prediction c^j is handled in the last layer without activation and an output dimension of 3.

The *decomposition network* uses the output from N_{ϕ_1} and directly predicts the direct color d and the density in a layer without activation and 4 output dimensions (RGB+σ). The main network output is then passed to several ReLU activated layers which handle the BRDF compression N_{θ_2}. The feature outputs are as followed: 32, 16, 2 (no activation), 16, 16, 5 (no activation). The final five output dimensions correspond to the number of parameters of the BRDF model. The compressed embedding with two feature outputs is regularized with a $\mathcal{L}^2$ norm with a scale of 0.1 and further clipped to a value range of -40 to 40 to keep the value ranges in the beginning stable.

Per batch, 768 rays are cast into a single scene.

Loss and learning rate scheduling. For adjusting the losses and learning rate during the training, we use exponential decay: $p(i; v, r, s) = vr^{\frac{i}{s}}$. The learning rate then uses $p(i; 0.000375, 0.1, 250000)$, the direct color d loss is faded out using $p(i; 1, 0.75, 1500)$ and the alpha loss is faded in using $p(i; 1, 0.9, 5000)$. During the first 1000 steps, we also do not optimize the SGs parameters and first use the white balancing only to adjust the mean environment strength, as this step also sets the illumination strength per image based on the exposure values.

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.

We use the following four general steps to extract textured meshes:

1. A very dense point cloud representation of the surface is extracted. This step utilizes the same rendering functions used during training, which ensures that the resulting 3D coordinates are consistent with the training. To generate the rays for the rendering step, we sample the *decomposition network* for σ in a regular grid within the view volume determined by the view frustums of the cameras. We construct a discrete PDF from this grid, which is then sampled to generate about 10 million points where σ is high. The rays are constructed by following the normals at those points to get the ray-origins. We use the slightly jittered inverted normals as ray directions. See Fig. 8 a to c for visualizations of this step.

2. For meshing, we use the Open3D [73] implementation of the Poisson surface reconstruction algorithm [29] using the normals from NeRD. Before meshing, we perform two cleanup steps: First, we reject all points where the accumulated opacity along a ray is lower than 0.98. Secondly, we perform statistical outlier removal from Open3D. Those steps are visualized in Fig. 8 d and e

3. We use Blender's [17] *Smart UV Project* to get a simple UV-unwrapping for the mesh. Reducing the mesh resolution beforehand is an optional step which reduces the computational burden for using the mesh later. This is also done using Blender via *Decimate Geometry* or the *Voxel Remesher*.

4. We bake the surface coordinates and geometry normals into a floating-point texture of the desired resolution. The textures are generated by generating and rendering one ray per texel to look up the BRDF parameters and shading normals with NeRD. A result is show in Fig. 8 f.

Dataset details. In Table 2, we list the trained resolution, the number of total images, and the test train split for each dataset.

Dataset	Resolution (W×H)	#Images	#Train	#Test
Globe	400×400	210	200	10
Car Wreck	400×400	210	200	10
Chair	400×400	210	200	10
Ethiopian Head	500×500	66	62	4
Gold Cape	456×456	119	111	8
Gnome	752×502	103	96	7

Table 2: **Dataset Overivew.** Overview about the resolution and number of images used for training.

Results. In this section, we show more visual and qualitative results for our training scenes. First, we show the performance on our other real world datasets (Gold Cape and Ethiopian Head). Samples are shown in Fig. 9. The details are preserved and apparent in our reconstructions. The

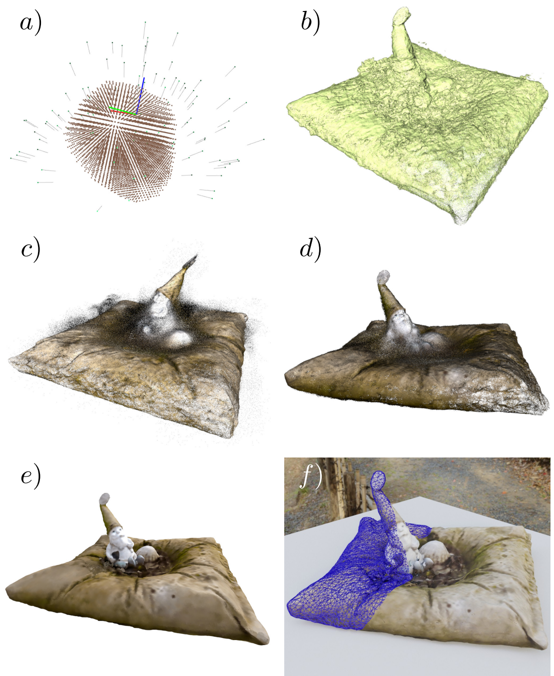


Figure 8: **Mesh Generation.** a) Cameras and view frustum. b) Points sampled where sigma was high. c) Rendered pointcloud with basecolor. d) Outlier removal. e) Mesh with vertex-colors. f) Low res mesh with material textures.

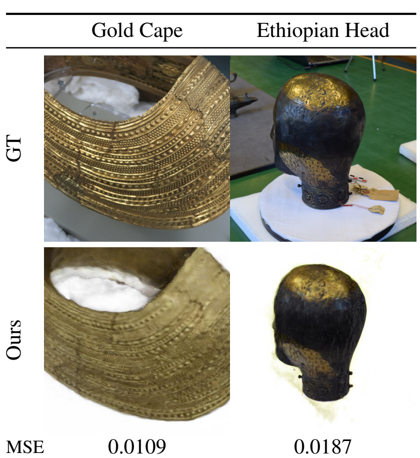


Figure 9: **Real World Novel View Synthesis.** Comparison on real-world samples with novel view synthesis on test dataset views.

reflective properties also match closely. We also want to highlight the prediction quality compared to NeRF [46] in Fig. 10. Here, especially in the scenes with varying illumination (Chair and Gnome), NeRF fails as expected. Our method decomposes the information, and after rendering the view synthesis is close to ground truth. In the scenes

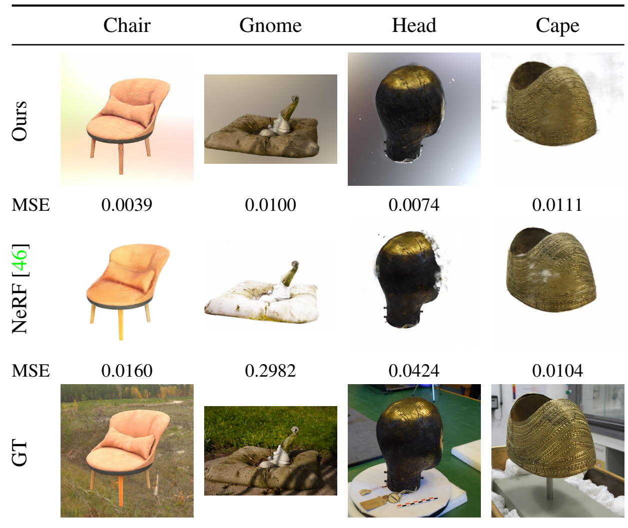


Figure 10: **Comparison with NeRF.** Comparison with NeRF on various scenes. Here, it is evident that NeRF fails as expected on scenes with varying illumination (Chair, Gnome).

with fixed illumination (Head and Cape), the performance between NeRF and our method is on par in most parts. The main difference in MSE is due to the baked-in highlights of NeRF. Our physically grounded design using rendering reduces these artifacts drastically. We also want to point out that relighting a scene is not possible in NeRF. Lastly, in Fig. 11 and Fig. 12 the accuracy in BRDF prediction is shown. In the globe scene (Fig. 11), the BRDFs do not reproduce the input perfectly. However, the re-rendering shows a small error and is visually also close. As the optimization is fully unconstrained, the decomposition found a solution which perfectly explains the input images. The chair scene (Fig. 12) captures the BRDF parameters and shape well, and the re-rendering MSE is extremely low. Visually, the ground truth and rendering are not distinguishable.

Notations. All notations in this work are listed in Table 3.

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. 3

[3] Rachel Albert, Dorian Yao Chan, Dan B. Goldman, and James F. O'Brian. Approximate svBRDF estimation from

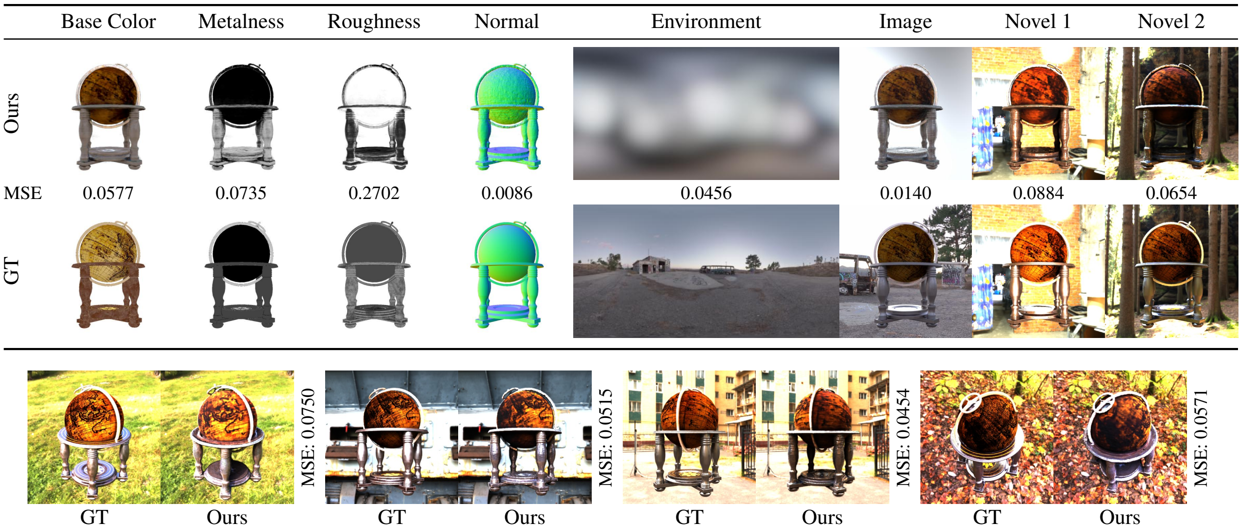


Figure 11: **Globe Decomposition.** Results on the decomposition of the synthetic globe scene. Even if the BRDF parameters do not capture the ground truth perfectly, the visual and quantitative error in re-rendering is extremely low. For most images, an alternative decomposition, which explains the input images, is found. As no other constraints exist, the solution is also plausible.

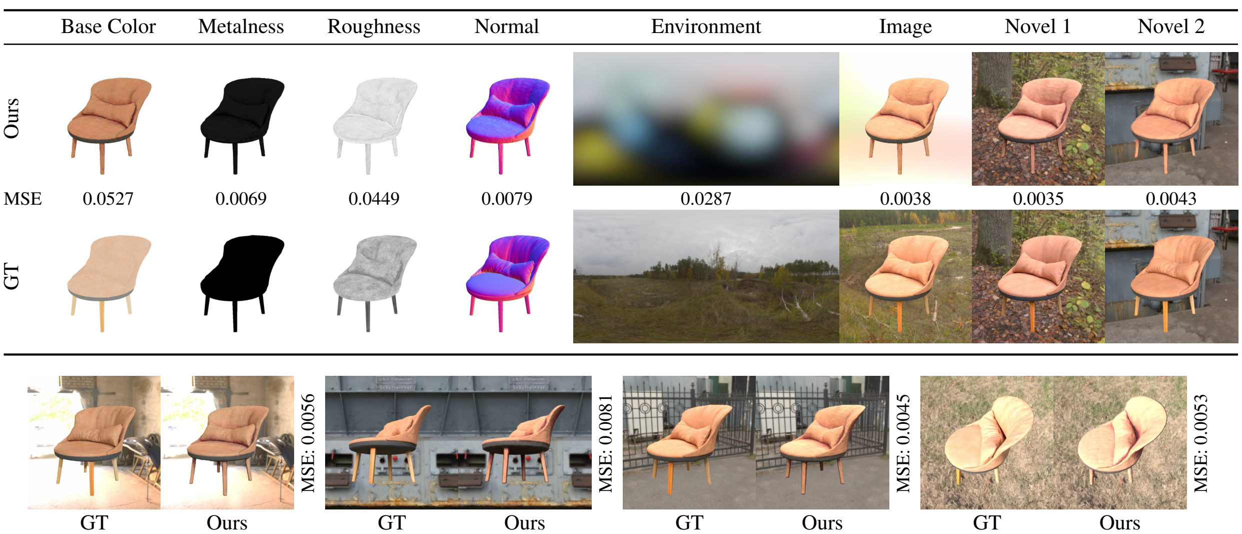


Figure 12: **Chair Decomposition.** Results on the decomposition of the synthetic chair scene. Here, a near perfect decomposition in shape and BRDF is found. All parameters are close to the ground truth and the error after rendering is extremely low. Visually, they capture the GT nearly perfectly.

mobile phone video. In *Eurographics Symposium on Rendering*, 2018. 3

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

[5] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[6] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015. 1, 3

[7] H.G. Barrow and Joan M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978. 3

[8] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. In *ACM Transactions on Graphics (SIGGRAPH)*, 2014. 3

[9] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall,

Symbol	$\in$	Description
q	$\mathbb{N}$	Number of images
s	$\mathbb{N}$	Number of pixel in each image
I_j	$\mathbb{R}^{s\times 3}; j \in 1, ..., q$	A specific image
$\boldsymbol{x}$	$\mathbb{R}^3$	The 3D coordinate in (x,y,z)
$\boldsymbol{b}$	$\mathbb{R}^5$	The BRDF parameters for the analytical cook torrance model
$\boldsymbol{n}$	$\mathbb{R}^3$	The surface normal
σ	$\mathbb{R}$	The density in the volume
$\boldsymbol{\Gamma}$	$\mathbb{R}^{24\times 7}$	The parameters for the spherical Gaussians environment illumination
t	$\mathbb{R}$	Used to query a position in distance t on a ray
$\boldsymbol{o}$	$\mathbb{R}^3$	The ray origin
$\boldsymbol{d}$	$\mathbb{R}^3$	The ray direction
$\boldsymbol{p}$	$\mathbb{R}^z$	A placeholder for a output of an object. Can be either BRDF parameters $\boldsymbol{b}$ or color $\boldsymbol{c}$. The dimensions z are dependent on the output type.
$\boldsymbol{c}^j$	$\mathbb{R}^3$	The potentially illumination dependent optimized color for the image j
$\boldsymbol{c}^j_{\omega_o r}$	$\mathbb{R}^3$	The illumination and view dependent optimized color for the image j
$\hat{\boldsymbol{c}}^j$	$\mathbb{R}^3$	The actual color for the image j
t_n	$\mathbb{R}$	The near clipping distance for the view frustum
t_f	$\mathbb{R}$	The far clipping distance of the view frustum
$\boldsymbol{\omega_i}$	$\mathbb{R}^3$	The incoming light direction (Pointing away from the surface)
$\boldsymbol{\omega_o}$	$\mathbb{R}^3$	The outgoing reflected light direction (Pointing away from the surface)
Ω		Defines the hemisphere at a point in normal direction $\boldsymbol{n}$
Function	$\in$	**Description**
$L_o(\boldsymbol{x}, \boldsymbol{\omega_o})$	$f(\mathbb{R}^3 \times \mathbb{R}^3) \mapsto \mathbb{R}^3$	The amount of outgoing light in the specified direction
$L_i(\boldsymbol{x}, \boldsymbol{\omega_i})$	$f(\mathbb{R}^3 \times \mathbb{R}^3) \mapsto \mathbb{R}^3$	The amount of incoming light from a specified direction
$f_r(\boldsymbol{x}, \boldsymbol{\omega_i}, \boldsymbol{\omega_o})$	$f(\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3) \mapsto \mathbb{R}^3$	The BRDF, which is dependent on the position on the surface and the incoming and outgoing light directions
$\rho_d(\boldsymbol{\omega_o}, \boldsymbol{\Gamma}, \boldsymbol{n}, \boldsymbol{b})$	$f((\mathbb{R}^3 \times \mathbb{R}^7 \times \mathbb{R}^3 \times \mathbb{R}^5) \mapsto \mathbb{R}^3$	The diffuse lobe evaluation using spherical Gaussian representations
$\rho_s(\boldsymbol{\omega_o}, \boldsymbol{\Gamma}, \boldsymbol{n}, \boldsymbol{b})$	$f((\mathbb{R}^3 \times \mathbb{R}^7 \times \mathbb{R}^3 \times \mathbb{R}^5) \mapsto \mathbb{R}^3$	The specular lobe evaluation using spherical Gaussian representations
$\gamma(\boldsymbol{x})$	$f(\mathbb{R}^3) \mapsto \mathbb{R}^{3z}$	Maps the input point coordinate to a higher dimensional embedding using z Fourier embedding

Table 3: **Notations.** Overview of all notations used in this work.

Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *ArXiv e-prints*, 2020. 2, 3, 5

[10] 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, 3

[11] 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. 2, 3

[12] Mark Boss, Fabian Groh, Sebastian Herholz, and Hendrik P. A. Lensch. Deep Dual Loss BRDF Parameter Estimation. In *Workshop on Material Appearance Modeling*, 2018. 2, 3

[13] 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, 3, 5

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

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

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

[17] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 9

[18] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1982. 4, 5

[19] 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. 3

[20] 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. 3

[21] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. In *Eurographics Symposium on Rendering*, 2020. 3

[22] 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. 3

[23] 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. 2, 3

[24] 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

[25] 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

[26] 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

[27] Mohamed Hachama, Bernard Ghanem, and Peter Wonka. Intrinsic scene decomposition from rgb-d images. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

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

[29] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 9

[30] 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

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ArXiv e-prints*, 2014. 6

[32] Pierre-Yves Laffont and Jean-Charles Bazin. Intrinsic decomposition of image sequences from local temporal variations. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[33] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient brdf importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 2004. 2, 3

[34] Hendrik Lensch, Jan Kautz, Michael Gosele, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Conference on Rendering*, 2001. 2, 3

[35] Hendrik P.A. Lensch, Jochen Lang, M. Sa Asla, and Hans-Peter Seidel. Planned sampling of spatially varying brdfs. In *Computer Graphics Forum*, 2003. 2, 3

[36] L. Lettry, K. Vanhoey, and L. Van Gool. DARN: a deep adversarial residual network for intrinsic image decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, 2018. 3

[37] 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. 3

[38] 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

[39] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[40] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[41] 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. 3

[42] 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

[43] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[44] 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

[45] 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

[46] 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, 5, 7, 10

[47] 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. 2, 3

[48] Takuya Narihira, Michael Maire, and Stella X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[49] Thomas Nestmeyer and Peter V Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

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

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

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

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

[54] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *European Conference on Computer Vision (ECCV)*, 2020. 3

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

[56] 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

[57] 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. 3

[58] Jian Shi, Yue Dong, Hao Su, and Stella X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[59] 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

[60] 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

[61] 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

[62] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005. 3

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

[64] 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, 5

[65] 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. 3

[66] 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

[67] 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

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

[69] Genzhi Ye, Elena Garces, Yebin Liu, Qjonghai Dai, and Diego Gutierrez. Intrinsic video and applications. In *ACM Transactions on Graphics (ToG)*, 2014. 3

[70] 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. 3

[71] 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. 3

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

[73] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *ArXiv e-prints*, 2018. 9

[74] Tinghui Zhou, Philipp Krähenbühl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3