Scene Decomposition and Relighting from Image Collections in Neural Rendering

Dongqing Wang

Abstract—**The focus of our research is to generate controllable photo-realistic images of real-world scenes from existing observations, i.e., the inverse rendering problem. The approaches we focus on are those through neural rendering, utilizing neural network to decompose the scene, learn its physical properties and render with novel lighting condition. In this proposal, we discuss three papers and how they relate to our research topic. We first look at a simple framework representing 3D scenes as volumetric radiance field for view synthesis; Then we look at a modification of the first paper to allow scene decomposition for illumination, geometry, surface reflectance, etc., for relighting; we lastly present a method using signed distance functions (SDF) for scene geometry addressing drawback of previous methods. Finally, we discuss our proposed solution for the problem and possible future research directions.**

Index Terms—**Physics-Based Rendering, Inverse Rendering, Neural Rendering, Scene Relighting.**

I. Introduction

The **inverse rendering problem**, i.e., recovering an object's shape, material and environment illumination from images such that arbitrary views can be rendered with novel lighting condition, has long been deemed as a highly ill-posed problem in graphics and vision community due to multiple times of interaction between light and the scene before reaching the observer [1–4].

Traditionally researchers started from physics perspective and explicitly modeled scene geometry and properties. They either control lighting and camera view setting with complex capturing system [5, 6], or estimate surface reflectance and scattering under strict laboratory conditions [7–9]. These methods produce physically accurate results but are hard to use outside lab setting and are error-prone. Image-based techniques render new observations directly out of 2D image, but require strong priors such as assumption of known 3D geometry [10]. They also often suffer from artifacts due to inaccurate interpolation of input views, and heavily relies on training data.

Recently, Neural Rendering has advanced in data-driven solutions for the problem. It combines the advantage of explicit modeling and image-based methods and constructs a physics-based 3D representation of the scene with neural fields to be queried for rendering.The state-of-the-art **neural field scene representation** attempt to estimate scene geometry and jointly recover fully-factorized 3D relightable model of the scene using existing image observations of the scene [1, 11–14]. A typical neural field algorithm is given in Figure.1 [4]. We feed the sampled coordinates into scene representation represented by neural networks, query scene properties to be evaluated as output to calculate reconstruction loss for training.

There are two major neural scene representations in the context of inverse rendering, volumetric radiance field [15] and signed distance function (abbr. SDF) [14]. Note that SDF representation encodes scene geometry as the zero-level set of the SDF modeled with MLP, while radiance field encodes volume and radiance (or its factorization) in the same network.

II. Background

Regardless of the scene representation, systems rendering scene under novel light condition must approximate the rendering equation using a specified surface reflectance model. Here we provide a list of background references:

1). The **rendering equation** in [16] describes the radiance leaving a point $\mathbf{x}$ through direction $\boldsymbol{\omega_o}$ as the sum of the radiance that $\mathbf{x}$ emits in direction $\boldsymbol{\omega_o}$ and the reflected/scattered radiance at $\mathbf{x}$ in the outgoing direction $\boldsymbol{\omega_o}$ received from all incoming directions $\boldsymbol{\omega_i}$:

$$L(\boldsymbol{\omega_o}, \mathbf{x}) = L_e(\boldsymbol{\omega_o}, \mathbf{x}) + \int_\Omega L(\boldsymbol{\omega_i}, \mathbf{x}') f_r(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}; \mathbf{x})(\boldsymbol{\omega_i} \cdot \boldsymbol{n}) d\boldsymbol{\omega_i}, \tag{1}$$

in which Ω represents the domain of $\omega_\mathbf{i}$, $f_r(.)$ denotes the bidirectional scattering function, which computes the incoming radiance at $\mathbf{x}$ along $\omega_\mathbf{i}$ is reflected along $\omega_\mathbf{o}$, $\mathbf{n}$ denotes surface normal at $\mathbf{x}$.

2). The general **Cook-Torrance/Microfacet** shading model detailed in [17] is used for modeling specular reflectance in this proposal. It is based on the idea that rough surfaces can be seen as a collection of microfacets, and defines the specular reflection on an opaque surface, and the works we present make use of its simplified version defined in [18]:

$$f_r(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}; \mathbf{x}) = \frac{\mathcal{D}(\mathbf{h})\mathcal{F}(\boldsymbol{\omega_i}, \mathbf{h})\mathcal{G}(\boldsymbol{\omega_i}, \boldsymbol{\omega_o}, \mathbf{h})}{4(\mathbf{n} \cdot \boldsymbol{\omega_i})(\mathbf{n} \cdot \boldsymbol{\omega_o})}, \tag{2}$$

in which $\mathbf{h} = \frac{\boldsymbol{\omega_o} + \boldsymbol{\omega_i}}{|\boldsymbol{\omega_o} + \boldsymbol{\omega_0}|}$, $\mathcal{D}(.)$ denotes the normal distribution function. $\mathcal{F}(.)$ is the Fresnel reflection coefficient and $\mathcal{G}(.)$ is the geometric attenuation, both vary w.r.t materials.

3). Used to represent direct or indirect light sources in this proposal, the **spherical Gaussian** function has the form [19]:

$$G(\mathbf{v}; \xi, \lambda, \mu) = \mu e^{\lambda(\mathbf{v} \cdot \xi - 1)}, \tag{3}$$

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

In this proposal, we focus on three recent works that are crucial in the area. The first paper [11] opens up a line of neural rendering research combining neural implicit functions together with rendering technique to achieve photo-realistic rendering results [20], and serves as a simplistic starting

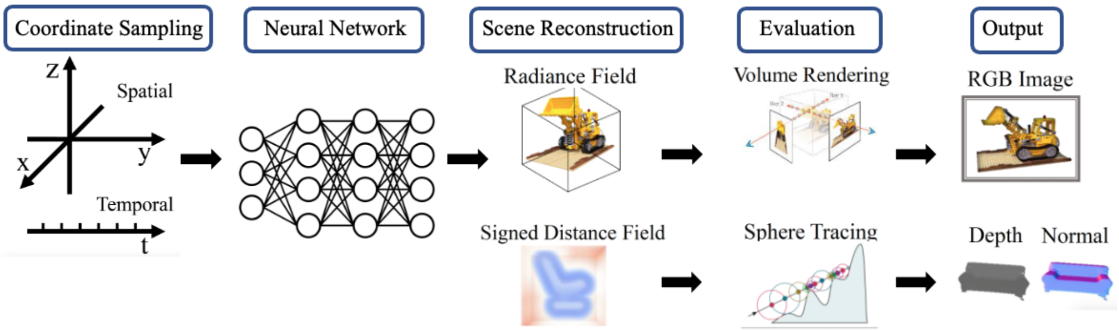

Fig. 1. Pipeline for a typical neural field algorithm [4]. First, sampled coordinates are fed into scene representation for field quantities, which in our context would be scene properties. Then the quantities as evaluated to calculate reconstruction loss.

framework for our work. The second paper [12] modifies NeRF to allow scene intrinsic factorization and rendering for relighting synthesis. We then evaluate the drawbacks of methods following volumetric scene representation. In presenting the third paper [13] we look at the advantages of using SDF for scene geometry estimation. Lastly, we present our attempt at the problem so far, and discuss future research plans.

III. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

A. The problem

This paper works on novel view synthesis for 3D scenes. It represents 3D scenes implicitly through a coordinate-based multilayer perceptron by encoding volume density and radiance at each 3D location. The scene volume is reconstructed through 2D image observations of the scene. Previous works [21, 22] also proposed using neural network as implicit scene representation, but only work with simple synthetic data, and are either not fully continuous or require prohibitively complex setup for training. By adding **positional encoding** to input coordinate samples and using hierarchical sampling technique, NeRF is able to produce high-quality novel view for complex real world scenes.

NeRF propose a data-driven solution for view synthesis by modeling 3D scene as an emissive volume without scattering of light encoded within a radiance field. It uses a classical volume rendering technique [23] that is trivially differentiable. It leads a line of following research in neural rendering.

B. Method

1) Setup: NeRF represents a continuous scene as a 5D vector-valued function $F : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, which takes in input of a 3D location $\mathbf{x} = (x, y, z)$ and a 2D viewing direction $\mathbf{d} = (\theta, \phi)$ as 3D Cartesian unit vector, and outputs an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ, see fig.2. The 5D scene representation F is approximated with an MLP network F_Θ whose weights Θ NeRF aim to optimize. The representation is multi-view consistent by restricting σ as only a function of $\mathbf{x}$, while predicting $\mathbf{c}$ as a function of both location and viewing direction, which enables view-dependent effects such as specular highlights..

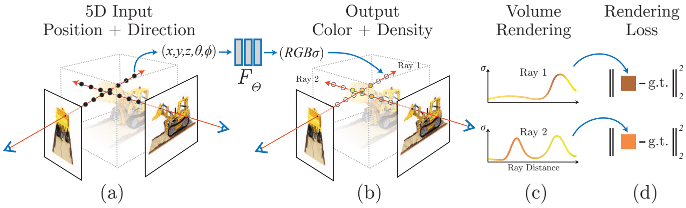

Fig. 2. An overview of NeRF scene representation and differentiable rendering procedure reproduced from [11]. (a) Sampling 5D coordinates along camera rays, (b) feeding those locations into an MLP to produce a color and volume density, (c) using volume rendering techniques for color compositing. (d) compare predition to ground truth image for loss [11]

2) Volume Rendering for ray evaluation: At any point in space, the scene is represented as the volume density $\sigma(\mathbf{x})$ and directional emitted radiance $\mathbf{c}(\mathbf{x}, \mathbf{d})$. NeRF interprets the volume density as the differential probability of a ray terminating at an infinitesimal particle at location $\mathbf{x}$. To render a view, NeRF calculates the expected color $C(\mathbf{r})$ for camera ray $\mathbf{r}(t) = \mathbf{o}+t\mathbf{d}$ with near and far bound t_n, t_f traced through each pixel of desired virtual camera by:

$$\begin{aligned} C(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \\ \text{where} \quad T(t) &= \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds) \end{aligned} \tag{4}$$

denotes the accumulated transmittance along the ray from t_n to t. This is estimated by applying stratified sampling approach on ray marching, where $[t_n,\ t_f]$ is partitioned in 64 evenly-spaced bins from each they draw on sample uniformly at random: $t_i \sim \mathcal{U}[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)]$. $C(\mathbf{r})$ can then by approximated by:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{c}_i, \text{ where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j)$$

$\delta_i = t_{i+1} - t_i$ measures distance between adjacent samples.

3) Optimization improvements: To achieve state-of-the-art quality, the model introduces two additional improvements.

Positional Encoding of input coordinates facilitate deep networks to learn high frequency functions. As shown in

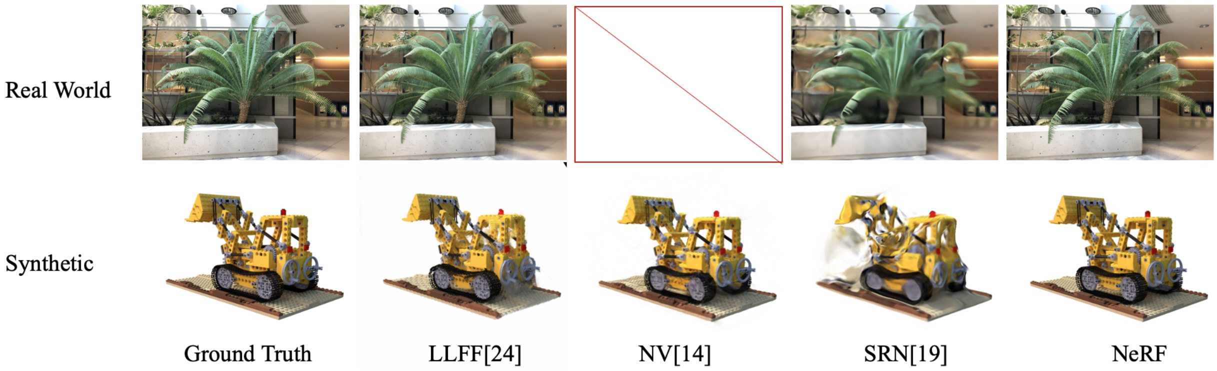

Fig. 3. Comparison for test-set views on selection of real world scene and synthetic scene across different methods. NeRF is able to recover fine detail in both geometry and appearance. On synthetic Lego, LLFF exhibits band ghosting artifact inside the object; NV cannot capture fine details; SRN produces distorted rendering. On real world fern, NV fails in this case; LLFF is designed for this use and therefore performs well, while NeRF produce more consistent geometry across views; SRN captures only low-frequency details of the object.

[24, 25], passing input coordinates through Fourier feature mapping avoids the bias of MLP towards low frequency functions, and improves the performance of coordinate-based MLPs in reconstructing finer scene details.

Specifically, NeRF maps three coordinate values in $\mathbf{x}$ (x, y, and z components range from 1 to 1) and three components of unit viewing direction $\mathbf{d}$ separately with $\gamma(x) = (\sin(2^0\pi x), cos(2^0\pi x), ... \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x))$. We take $L = 10$ for $\mathbf{x}$ and $L = 4$ for $\mathbf{d}$.

Hierarchical volume sampling improves the inefficiency of stratified sampling with evenly-divided bins in free space and occluded regions which do not contribute to rendered image. The scene representation utilizes a “coarse” network evaluated at locations sampled with stratified sampling along each ray. Using output from coarse network, a separate “fine” network is evaluated at samples locations biased towards locations with higher estimated volume density through inverse transform sampling, and therefore accumulates weighted color samples.

4) Training: Around 200 captured RGB images are required to train each scene. At the training stage, the model samples from the set of all pixels (each with one ray) and uses hierarchical sampling for coarse and fine network. Then the model predicts pixel-wise color with Eqn. (4). The loss is calculated via MSE between rendered and true pixel colors for both coarse and fine renderings.

C. Results

In the article, both synthetic rendering of objects and real world captured images are used for evaluating the performance of this method comparing to other state-of-the-art methods Neural Volume [15], Local Light Field Fusion [26] and Scene Representation Network [21].

Quantitative results On synthetic datasets rendered with complex object with non-Lambertian materials by Blender, NeRF is able to outperform all SOTA baseline on PSNR, SSIM and LPIPS [27] metrics. As for front-facing capture of real scenes, Neural Volume [15] is unable to train on this scenario. NeRF is able to outperform the other two baseline methods on PSNR and SSIM, but is not able to beat LLFF as the latter is specifically designed for dataset captured in the particular setup.

Qualitative results Examples are in Fig.3. For the synthetic case, NeRF is able to recover fine details for both geometry and material while preserving 3D view consistency. LLFF shows band artifacts and ghosting effects on certain cases; SRN produces blurry and distorted renderings, and NV fails to capture finer geometric details. On real captured scenes NV fails completely; SRN only captures color variance, and is unable to produce any fine details; LLFF is designed for this type of data, but NeRF is still able to achieve higher view consistency over certain cases.

D. Discussion

NeRF presents the first continuous scene representation to render from real captured RGB images and synthesize photo-realistic novel views. Compare to its baseline methods, NeRF is advantageous not only in that it achieves superior image quality and better metrics, but also for its simplicity.

NeRF provides the foundation of the proposal, offering many valuable insights. 1) It demonstrates that encoding 3D volume in a continuous MLP and solves the dilemma of limited resolution for voxels is possible, unlike claimed by Neural Volume to be impossible due to limited size of the network. 2) The volume density is only location dependent while radiance is also view dependent, creating asymmetry between $\mathbf{x}$ and $\mathbf{d}$, which contribute to avoiding *shape-radiance entanglement* [28]. 3) The differentiable nature of volume rendering allows the network parameters to be optimized directly by using gradient descent to avoid discontinuity issue in traditional graphics approach.

There are also striking limitations to this method. NeRF requires training time in terms of days for a single scene and around 30s for inferring a single view, and it does not allow animated motion synthesis of the object, performs poorly for view interpolation, etc. Most importantly for our case, NeRF does not consider varying illumination in the scene because the volume emission is baked in as direct output of the MLP, and does not model indirect illumination for dynamic rendering of shadows. In the following section, we look at a work that takes advantage of the simple skeleton framework provided by NeRF, and provide insights on how to enable relighting for inverse rendering problem.

IV. NeRD: Neural Reflectance Decomposition from Image Collections

A. The problem

After NeRF proposed a working solution for implicit scene representation, NeRD modifies its model architecture for inverse rendering problem, i.e., estimating one or more illumination, reflectance properties and shape of a scene given its image collections captured under possibly varying illumination.

As NeRF assumes baked-in lighting within radiance field and does not work for varying illumination, NeRD modifies the learnt quantities within the radiance field from RGB colors to relightable reflectance parameters and surface normal prediction. It continues with implicit continuous scene representation in NeRF, and produces relit images under novel illumination as well as arbitrary views.

Works such as [1, 22] have focused on modeling light transport and supports shadow rendering and other advanced lighting effects with high computational cost as trade-off. But NeRD does not explicitly model shadow and does not model indirect illumination. It also assumes opaque surface and therefore does not consider Bidirectional Transmittance Distribution Function(BTDF), and assumes BRDF representing full surface reflectance.

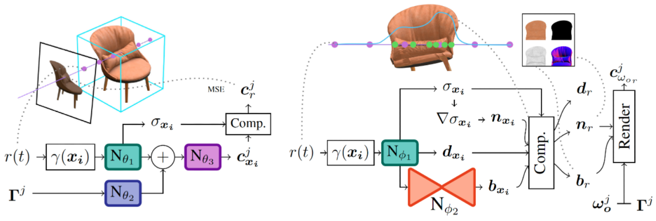

Fig. 4. The architecture of NeRD consists of two networks reproduced from [12]. Here, N_{ϕ_1}/N_{ϕ_2} denote instances of the main network encoding reflectance volume, γ(x) is the Fourier Embedding, and Γ_j denotes the SG parameters per image j. c is output color and σ is volume density. "Comp" denotes the alpha composition of individual samples along the ray [12].

B. Method

1) Overview: NeRD aims to optimize a 3D volume, in which at each location $\mathbf{x} = (x, y, z)$ it encodes volume density $\sigma \in \mathbb{R}$, and replace radiance value with BRDF parameters $\mathbf{b} \in \mathbb{R}^5$ and surface normal $\mathbf{n} \in \mathbb{R}^3$. A single environment map is estimated as the sum of 24 spherical Gaussian mixtures with $\Gamma \in \mathbb{R}^{24\times 7}$ as Eqn. (3).

2) Network Architecture & Training: **The Sampling Network** in the left of fig.4 resembles the "coarse" network of NeRF, but estimate a view-independent and illumination-dependent color at each location by concatenating the output of the encoding layers with a condensed embedding of spherical Gaussians inspired by NeRF-w [29]. It is optimized by MSE loss w.r.t. ground truth pixel color. The sampling network establish a sampling pattern for decomposition network.

The Decomposition Network in the right of fig.4 has a decomposition step and a rendering step between output of 3D volume and color prediction. The decomposition step estimates view and illumination dependent BRDF parameters based on the Disney BRDF basecolor-metallic parameterization [30] and surface normal at each point. NeRD then calculates specular component via Eqn.(2). Specifically, the model uses the sum of 24 spherical Gaussian evaluations to approximate Eqn.(1):

$$L_o(\boldsymbol{\omega_o}, \mathbf{x}) \approx \sum_{m=1}^{24} \rho_{sg}(\boldsymbol{\omega_o}, \Gamma_m, \mathbf{n}, \mathbf{b}), \tag{5}$$

in which L_o denotes emitted radiance, ρ_{sg} denotes the spherical Gaussian rendering evaluation, $\omega_\mathbf{o}$ denotes the outgoing ray direction. The viewing direction is the inverse ray direction.

In the decomposition step, NeRD does not directly estimate surface normals, instead defines it as the normalized negative gradient of the density field: $\mathbf{n} = -\frac{\nabla_\mathbf{x}\sigma}{\|\nabla_\mathbf{x}\sigma\|}$ inspired by [2]. To estimate BRDF, NeRD passes the output of 3D encoding volume to an auto-encoder which creates a 2-dimensional latent space encoding all possible spatially varying BRDF in the scene. A decoder then outputs the 5D BRDF parameter $\mathbf{b}$ based on basecolor-metallic decomposition.

The re-rendered result of evaluating randomly-generated rays through estimated volume density, normal and BRDF is compared with input images via Mean Squared Error loss, which is backpropagated to all scene intrinsic parameters including illumination Γ for joint optimization.

Tonemapping on color prediction is applied by auto-exposure post-processing directly before comparing with MSE loss to account for sRGB curve and white balancing applied to the Low Dynamic Range images of the dataset. It calculates the luminance and exposure value of input image, and applies white balance to predicted RGB based on the same exposure value.

3) Mesh Extraction: NeRD is able to extract textured mesh after training via a pipeline that generates a point cloud, computing a mesh and generating texture map for wrapping. The method is not highly efficient nor trivial to use, but it works with traditional graphics engine and produce relit result more easily. NeRF is only able to generate mesh with texture that has baked lighting.

C. Results

NeRD uses both synthetic scenes with ground truth BRDF to showcase the decomposition results and real world scenes captured under varying illumination.

1) Decomposition result: The output reflectance parameter estimated by NeRD is able to be used for rendering and provide an image that is very close to ground truth, although each part of decomposition does not necessarily match the ground truth decomposition. See Fig.5. NeRD runs relight model from [31] on NeRF to show that NeRF is unable to create coherent geometry and results in different BRDF maps across views. NeRD therefore claims that jointly optimizing

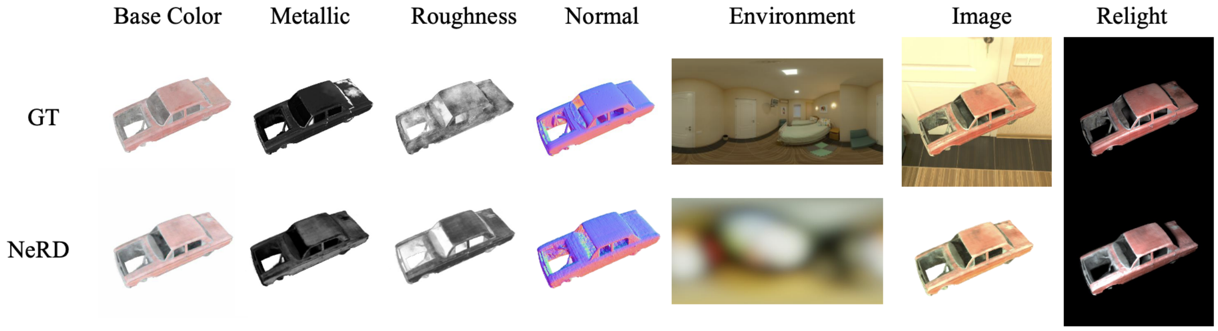

Fig. 5. Selected results on NeRD model decomposition and relight results. The environment map is reconstructed in low-frequency and loses details. The relit results are generated from ground truth lighting within a path tracer.

shape and reflectance parameters is necessary for scene intrinsic estimation.

2) Relighting and novel view synthesis: Relit result of NeRD is evaluated qualitatively, as stated by NeRD to be visually close to held-out validation images of the same scene. Several fine details are missing in the reconstruction, and specular highlight is missing occasionally from grazing angle. For novel view synthesis, NeRD uses a baseline NeRF-A inspired by [29] which works on dataset with varying illumination. NeRD is able to outperform both NeRF and NeRF-A quantitatively in the case with varying illumination in the dataset on both synthetic and real world scenes.

D. Discussion

NeRD proposes a method for estimating 3D shape, reflectance, illumination from RGB image collection based on coordinate-based implicit scene representation. It is closely related to NeRF, and optimizing scene intrinsics jointly through the quality of output rendering of the reconstructed scene representation. By enforcing the underlying shape and reflectance to be the same for one scene, the approach is able to converge and preserve consistency across various illumination condition.

For the aspect of the proposal, NeRD offers valuable insight for the problem we want to solve. NeRD shows that by altering quantities encoded in the 3D Neural volume, it is possible to perform scene editing, modifying scene illumination, material editing, etc. It gives a generic approach for reflectance estimation within the scene for later research to focus on improving specific materials such as glossy surface [3] It also leverages the link between surface normal and volume density, which however can also be a source of error.

Indeed when calculating surface normal directly from volume density, the accuracy of normals is strongly correlated to the quality of predicted volume density. Both NeRF and NeRD often produce noisy and low fidelity geometry, and is unable to adequately reconstruct fine details, especially for NeRD. Another issue is that even if we ignore the inaccurate assumption of NeRF that does not model light transport, NeRD decomposes BRDF down to a 2-D embedding space, and extract diffuse and specular encoding from the decoded embedding, causing in an entanglement between the diffuse and specular components of surface reflectance. Specular reflectance is modeled as a 3-channel RGB color contributing to surface reflectance in terms of ”specular tint”[30]. In next section, we discuss that modeling the geometry as a function of volume density using SDF resolves the first issue [20], and estimating specular component separately from diffuse improves the second issue.

V. PhysSG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting

A. The problem

Similar to the article discussed in previous section, PhysSG presents an end-to-end inverse rendering pipeline. While NeRD is a method that recovers surface reflectance and object shape jointly under varying illumination, it models BRDF parameters from decoded low dimensional latent space and computes both diffuse color and specular highlight jointly from basecolor and metallic parameter term and occasionally produces noisy surface normal quantities.

PhySG takes a step back and examines the scenario for dataset with fixed unknown illumination. It models surface reflectance in a physically-based manner by modeling specular highlight as spherical Gaussian functions taking same form as the approximation for environment illumination. For scene representation, PhySG uses level sets of neural networks as SDF to represent 3D shapes in the scene regularized by Eikonal regularization[14].

B. Method

1) Overview: The scene geometry and surface reflectance are jointly optimized in PhySG's model, but unlike NeRF and NeRD, PhySG models shape and surface-emitted radiance with separate network. For a camera ray $\mathbf{r} = \mathbf{o} + t\mathbf{d}$, its intersection $\mathbf{x}$ with scene surface represented with SDF is found via sphere tracing[33]. Spatially-varying diffuse component is represented via a MLP network mapping location to RGB color, and specular component is assumed to be constant and monochrome, represented with a SG.

2) Geometry Estimation: The geometry network of PhySG is based on Implicit Differentiable Renderer[2]. Geometry is presented as the zero level set of a MLP network:

$$S(\mathbf{x}, \Theta) = \{\mathbf{x} \in \mathbb{R}^3 | f(x, \Theta) = 0\}, \tag{6}$$

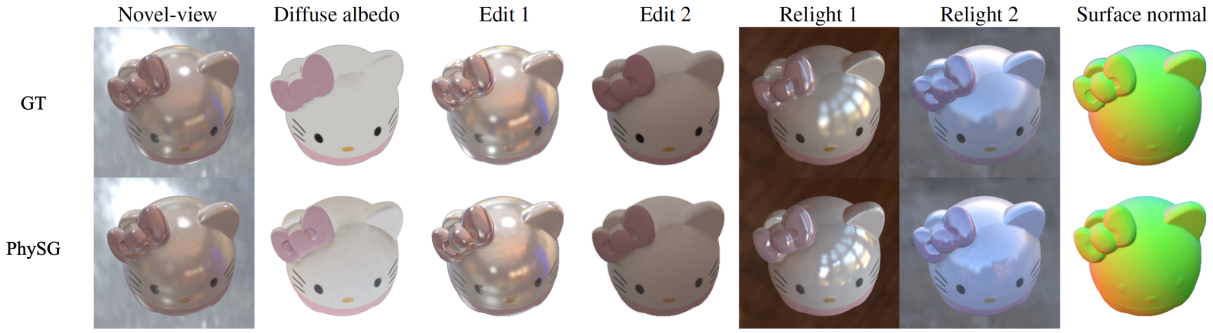

Fig. 6. Selected results of PhysG on synthetic data. For a novel test view, comparison is made between predicted RGB image, diffuse albedo, specular BRDF results and relighting results to ground truth images rendered by Mitsuba renderer [32]. PhysG notes the scale ambiguity in inverse rendering problems; hence it aligns estimated diffuse albedo to the ground truth. [13]

where $\mathbf{x}$ is a 3D location, f with parameter Θ is the MLP modeling the SDF to its zero level set $S(\mathbf{x},\Theta)$. Similar to NeRF, a 3D location point is encoded with positional encoding. The surface normal is given by the gradient of SDF $\mathbf{n} = \nabla_x S$. For optimization, gradients are backpropagated to both $\mathbf{x}$ and $\mathbf{n}$ as in [2].

3) Appearance Modeling: After obtaining a ray intersection at $\mathbf{x}$, PhysG approximates Eqn. (1):

Reflectance is modeled as a sum of spatially-varying diffuse component $\mathbf{a}(\mathbf{x},\mathbf{\Phi})$ mapping locations to RGB colors and isotropic monochrome specular component $f_r(\omega_\mathbf{0},\omega_\mathbf{i};\mathbf{x}) = \frac{\mathbf{a}}{\pi} + f_s(\omega_\mathbf{0},\omega_\mathbf{i};x)$ which follows the Microfacet model in Eqn. (2). $\mathcal{D}(.)$ is represented by a single SG parameterized by $\mathbf{h}$, $\mathbf{n}$ and $\boldsymbol{\omega_o}$, as well as roughness parameter and the remaining $\mathcal{M}_x$ is approximated to be constant [34].

Shadowing term $\boldsymbol{\omega_i} \cdot \mathbf{n}$ is represent with a spherical Gaussian function integrated over the whole sphere [35].

$$\boldsymbol{\omega}_i \cdot \mathbf{n} \approx G(\boldsymbol{\omega}_i; 0.0315, \mathbf{n}, 32.7080) - 31.7003 \qquad (7)$$

Lighting is assumed to be distant direct illumination represented by environment maps. The environment map $L_i(\omega_\mathbf{i})$ is thereby represented by the mixture of 128 SGs.

The full approximation for the rendering equation is thereby given by an approximation of hemispherical integral for spherical Gaussian functions and dot product in a closed form solution.[35] The products of spherical Gaussian functions result in another spherical Gaussian function, which then to be integrated over the sphere.

The radiance of the ray $\mathbf{r}$ is subsequently given by evaluating the rendering equation in closed form through diffuse albedo $\mathbf{a}(\mathbf{x},\mathbf{\Phi})$ at $\mathbf{x}$, surface normal $\mathbf{n}$, environment map $\{\xi_k, \lambda_k, \mu_k\}_{k=1}^{K=24}$, specular BRDF $\{\lambda, \mu\}$, and viewing direction direction $\omega_\mathbf{o}$. Finally, the predicted color is compared to ground truth RGB, which is differentiable w.r.t the network parameter Θ for the SDF of zero-level set. See fig.7

4) Training: Apart from RGB loss which encourage closer radiance prediction, PhysG applies a mask loss encouraging more ray hit on object and incorporated the Eikonal regularization enforcing the MLP f to be approximately a SDF. The lobes of the environment map are initialized to distribute uniformly on the unit sphere using a spherical Fibonacci lattice[36], which is a fast algorithm for place nearly uniform point distributions on the unit sphere.

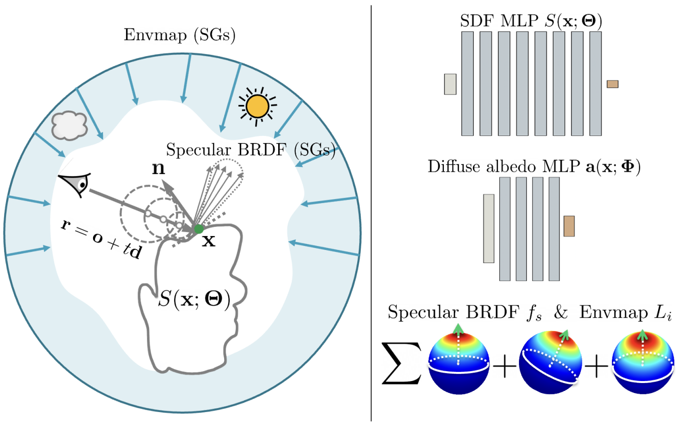
Envmap (SGs)
Specular BRDF (SGs)
n
$\mathbf{r} = \mathbf{o} + t\mathbf{d}$
x
$S(\mathbf{x};\mathbf{\Theta})$
SDF MLP $S(\mathbf{x};\mathbf{\Theta})$
Diffuse albedo MLP $\mathbf{a}(\mathbf{x};\mathbf{\Phi})$
Specular BRDF f_s & Envmap L_i
$\sum$ + +

Fig. 7. Pipeline reproduced from [13]. PhysG first uses sphere tracing to find the ray's intersection $\mathbf{x}$ with the geometry in the form of a SDF represented as an MLP S($\mathbf{x}$;Θ). The surface normal $\mathbf{n}$ at location $\mathbf{x}$ is then computed as SDF gradient. The spatially-varying diffuse albedo $\mathbf{a}$(x;Φ) is represented with an MLP. Given $\mathbf{n}$, $\mathbf{a}$, and viewing direction at $\mathbf{x}$, the SG renderer renders image to be compared with GT via image reconstruction loss to jointly optimize scene properties.

C. Results

PhysG performs experiments on both synthetic and real-world data limited to smooth object surfaces with specularities, i.e.rough conductor. To mitigate the inherent scale ambiguity between light and texture, PhysG applies a channel-wise scale factor to predicted image $\hat{I}$ to aligh with ground truth I. Specifically, let $\hat{I}_r$, I_r be the red channel of $\hat{I}$, I respectively. Then the scale factor s_r for the red channel is estimated via $s_r = \text{Median}(I_r/\hat{I}_r)$. This is similar to the auto-exposure post-processing in NeRD.

PhysG lists NeRF [11], IDR [2] and DVR [37] as baselines, while these methods all model surface color as baked-in radiance. NeRF does poorly on view extrapolation because it does not concentrate color around surfaces and cannot produce accurate specular reflection from surface. DVR fails to model glossy surfaces. IDR can model view-dependence and do view extrapolation but can't synthesize specular highlights due to lack of a physical model of appearance. PhySG performs well on the task as shown in fig.6.

D. Discussion

PhySG uses separate MLPs to reconstruct scene geometry and surface reflectance, and recovers environment illumination via mixture of spherical Gaussian functions with a setup that follows physically-based rendering more closely than NeRD.

For our proposal, PhySG provides insights for another possible implicit approach for neural rendering apart from radiance fields. PhySG gives a closed form solution for calculating the hemispherical integral for the product of incoming light and surface material properties in the context of representing geometry as SDF instead of radiance field. SDF as a zero-level set for neural network demonstrates higher level of explanability, as the intersection between ray and SDF as well as the surface normals given by exact solution [2] instead of an approximation as in radiance field. For next step of our research, we plan to focus on signed distance fields as primary scene representation.

There is a lot left to improve for the work still. Currently PhySG assumes full light source visibility with no self-occlusion. It works with an analytic BRDF instead of a data-driven one. It also assumes non-spatially-varying specular reflectance and monochrome specular highlight for the scene, and cannot recover lighting without additional supervision if the material is purely Lambertian. Finally, the training time for PhySG is on the order of days.

VI. Research Proposal

In this section, we first briefly present our experiments on the relighting problem with neural rendering approaches, and then discuss our vision for next step research directions.

1) Current Work: Our work is based on NeRD [12] discussed in Section IV. NeRD makes use of Disney BRDF parametrization and computes the diffuse and specular response based on its 2D BRDF embedding.

BRDF decomposition In our experiments, we look at shape-radiance entanglement and predict the diffuse and specular response of each 3D point. The renderer uses the Cook-Torrance analytical BRDF model. By changing the decomposition method of the scene, we aim to better disentangle surface color estimated by the neural volume and improve reconstruction quality. Specifically, we have the updated approximation of rendering equation as follows:

$$L_o(\boldsymbol{\omega_o}, \mathbf{x}) \approx \sum_{m=1}^{24} \rho_d(\boldsymbol{\omega_o}, \Gamma_m, \mathbf{n}, \mathbf{b}) + \rho_s(\boldsymbol{\omega_o}, \Gamma_m, \mathbf{n}, \mathbf{b}), \quad (8)$$

where ρ_d stands for the SG renderer of diffuse component and ρ_s is for specular component. Our model predicts more accurate roughness parameter than NeRD compares to ground truth, as shown in the third column of Fig.10. Roughness controls diffuse and specular surface reflectance.

Relight synthesis The original NeRD extracts a textured mesh from its Base color/Metallic BRDF decomposition. Given no textured mesh can be extracted from our method, we render the closest approximate of relighted result under these SGs approximated environment maps. With `SGRenderer`, we replace the SG arrays in the differentiable renderer with ground truth SG approximation, and generated relighted results of the scene to compare.

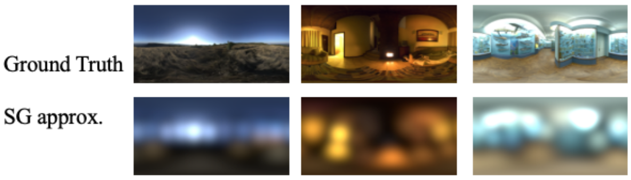

Fig. 8. Examples of environment maps approximated by 24 spherical Gaussian functions vs Ground Truth.

One observation for the approximated envmap is that SGs are not able to capture high radiance value (e.g. ≥ 1000 such as area near the sun) in the original HDR environment maps, and therefore results in the relighting synthesis is darker than ground truth envmap. In Fig.11, our results capture more highlight details than NeRD, despite the overall object being dark due to SGs error.

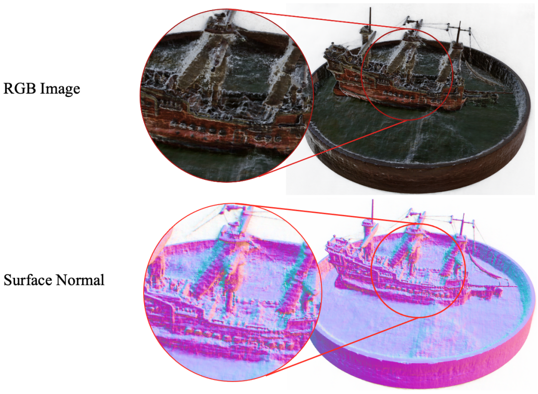

Fig. 9. Example of NeRD failing to capture the fine details of thin objects. Instead it produces noisy and low fidelity geometry, and is unable to adequately reconstruct fine details.

Through our experiments, we produce surface diffuse color that is less dependent on environment lighting, as opposed to the results produced by NeRD; we also produce better rendering in overcoming NeRD's hazy artifacts presented in the corner of the globe (see first column of fig.10). We realize the shortcoming of inferring the surface normal from opacity encoded in the volumetric radiance field. Estimating normal through defining it as the normalized negative gradient of the density field: $\mathbf{n} = -\frac{\Delta_x \sigma}{\|\Delta_x \sigma\|}$ as in NeRD provides an exact solution for SDF, but is merely a rough estimation of radiance field. We realize that using SDF for scene geometry representation provides higher accuracy for our model, as shown in the *Ship* test result as in fig.9.

We also realize a flaw in our assumption in which spherical Gaussiam form of illumination being considered as ground truth illumination. Using split-sum approximation [18] as in [38, 39] to precompute the integral of specular BSDF with

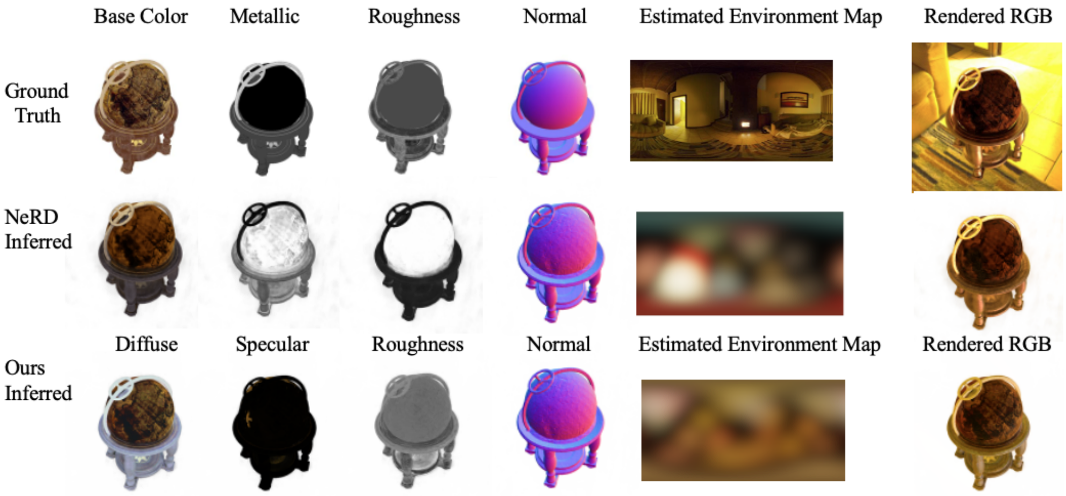

Fig. 10. Example of comparison of the decomposition results for each BRDF parameters between our methods, the NeRD model, and ground truth. Also included are estimated environment maps as well as rerendered RGB images comparing to reference images.

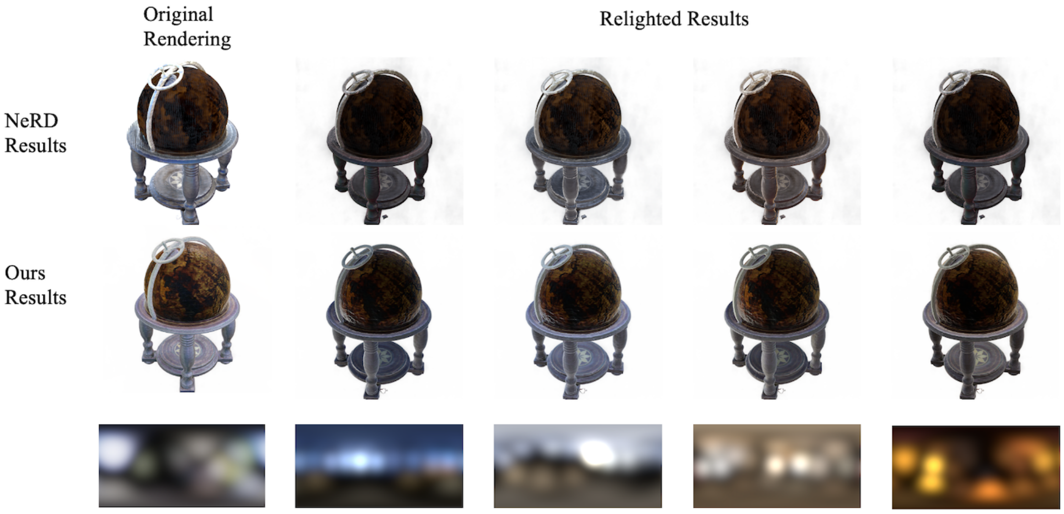

Fig. 11. Relighting synthesis for Globe dataset with approximated environment map. Our results are compared to previous method. From comparison we can see the specular component of our result is able to preserve texture details as opposed to previous method where the texture is too dark to be seen.

a solid white environment light into a 2D lookup texture increases the integration performance and provides finer details in environment illumination.

2) Future plans: Our first proposed direction is **Shape-radiance disentanglement** [28, 40]. As discussed in recent works on inverse rendering with neural approaches, real-time rendering techniques in graphics can provide insights on improving neural methods. For better environment illumination estimation, several works [38, 39, 41] uses split-sum approximation to precompute environment map factors; Neural Radiance Transfer Field [42] proposed using precomputed radiance transfer for limited light transport modeling with a differentiable path tracer, and thereby achieving 2-3 bounces of global illumination. We propose use to combine in our model mature computer graphics techniques such as PRT to estimate the scene parameters for more realistic and accurate scene relight synthesis.

We then propose **appearance acquisition** [43] for specific materials with neural methods. For instance, there are existing 3D volumetric models for specific type of materials, such as tracing light transport through volumetric hair [44]. Works have also been done optimizing glossy material via parametrization of outgoing radiance with surface normal and mirror reflection [3] to make specular appearance better-suited for interpolation. We believe that by looking into physics-based model of materials, we can reconstruct better material acquisition.

VII. Conclusion

This proposal is focused on inverse rendering problem with neural approaches. Through three selected papers, we summarize the advances and limitations of neural methods for the inverse rendering problem, discussed our proposed modifications on relighting and possible future research directions.

References

[1] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, "Neural reflectance fields for appearance acquisition," *arXiv preprint arXiv:2008.03824*, 2020.

[2] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.

[3] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," *arXiv preprint arXiv:2112.03907*, 2021.

[4] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.

[5] Y. Dong, G. Chen, P. Peers, J. Zhang, and X. Tong, "Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–12, 2014.

[6] R. Xia, Y. Dong, P. Peers, and X. Tong, "Recovering shape and spatially-varying surface reflectance under unknown illumination," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.

[7] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec, "Linear light source reflectometry," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 749–758, 2003.

[8] A. Ghosh, T. Chen, P. Peers, C. A. Wilson, and P. Debevec, "Estimating specular roughness and anisotropy from second order spherical gradient illumination," in *Computer Graphics Forum*, vol. 28, no. 4. Wiley Online Library, 2009, pp. 1161–1170.

[9] D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross, "Brdf representation and acquisition," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 625–650.

[10] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 2, pp. 234–257, 2003.

[11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[12] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 684–12 694.

[13] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," *arXiv preprint arXiv:2002.10099*, 2020.

[15] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 65:1–65:14, Jul. 2019. [Online]. Available: http://doi.acm.org/10.1145/3306346.3323020

[16] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 143–150.

[17] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces." *Rendering techniques*, vol. 2007, p. 18th, 2007.

[18] B. Karis and E. Games, "Real shading in unreal engine 4," *Proc. Physically Based Shading Theory Practice*, vol. 4, no. 3, p. 1, 2013.

[19] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," in *ACM SIGGRAPH Asia 2009 papers*, 2009, pp. 1–10.

[20] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4805–4815, 2021.

[21] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *Advances in Neural Information Processing Systems*, 2019.

[22] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7495–7504.

[23] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[24] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5301–5310.

[25] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *NeurIPS*, 2020.

[26] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, 2019.

[27] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[28] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv:2010.07492*, 2020.

[29] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[30] B. Burley and W. D. A. Studios, "Physically-based shading at disney," in *ACM SIGGRAPH*, vol. 2012. vol. 2012, 2012, pp. 1–7.

[31] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Learning to reconstruct shape and spatially-varying reflectance from a single image," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 269.

[32] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.

[33] J. C. Hart, "Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces," *The Visual Computer*, vol. 12, no. 10, pp. 527–545, 1996.

[34] A. Ngan, F. Durand, and W. Matusik, "Experimental analysis of brdf models," vol. 2, 01 2005, pp. 117–126.

[35] J. Meder and B. Brüderlin, "Hemispherical gaussians for accurate light integration," in *International Conference on Computer Vision and Graphics*. Springer, 2018, pp. 3–15.

[36] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, "Spherical fibonacci mapping," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–7, 2015.

[37] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler, "Extracting triangular 3d models, materials, and lighting from images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8280–8290.

[39] J. Hasselgren, N. Hofmann, and J. Munkberg, "Shape, light & material decomposition from images using monte carlo rendering and denoising," *arXiv preprint arXiv:2206.03380*, 2022.

[40] S. R. Marschner, *Inverse rendering for computer graphics*. Cornell University, 1998.

[41] M. Boss, A. Engelhardt, A. Kar, Y. Li, D. Sun, J. T. Barron, H. P. Lensch, and V. Jampani, "SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections," in *ArXiv e-prints*, 2022.

[42] L. Lyu, A. Tewari, T. Leimkuehler, M. Habermann, and C. Theobalt, "Neural radiance transfer fields for relightable novel-view synthesis with global illumination," *arXiv preprint arXiv:2207.13607*, 2022.

[43] E. Veach, *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[44] J. T. Moon, B. Walter, and S. Marschner, "Efficient multiple scattering in hair using spherical harmonics," in *ACM SIGGRAPH 2008 papers*, 2008, pp. 1–7.