Deep Reflectance Volumes: Relightable Reconstructions from Multi-View Photometric Images

Sai Bi[1], Zexiang Xu[1,2], Kalyan Sunkavalli[2], Miloš Hašan[2], Yannick Hold-Geoffroy[2], David Kriegman[1], Ravi Ramamoorthi[1]

[1] University of California, San Diego
[2] Adobe Research

Abstract. We present a deep learning approach to reconstruct scene appearance from unstructured images captured under collocated point lighting. At the heart of Deep Reflectance Volumes is a novel volumetric scene representation consisting of opacity, surface normal and reflectance voxel grids. We present a novel physically-based differentiable volume ray marching framework to render these scene volumes under arbitrary viewpoint and lighting. This allows us to optimize the scene volumes to minimize the error between their rendered images and the captured images. Our method is able to reconstruct real scenes with challenging non-Lambertian reflectance and complex geometry with occlusions and shadowing. Moreover, it accurately generalizes to *novel* viewpoints and lighting, including non-collocated lighting, rendering photorealistic images that are significantly better than state-of-the-art mesh-based methods. We also show that our learned reflectance volumes are editable, allowing for modifying the materials of the captured scenes.

Keywords: View synthesis, relighting, appearance acquisition, neural rendering

1 Introduction

Capturing a real scene and re-rendering it under novel lighting conditions and viewpoints is one of the core challenges in computer vision and graphics. This is classically done by reconstructing the 3D scene geometry, typically in the form of a mesh, and computing per-vertex colors or reflectance parameters, to support arbitrary re-rendering. However, 3D reconstruction methods like multi-view stereo are prone to errors in textureless and non-Lambertian regions [37, 47], and accurate reflectance acquisition usually requires dense, calibrated capture using sophisticated devices [5, 55].

Recent works have proposed learning-based approaches to capture scene appearance. One class of methods use surface-based representations [15, 20] but are restricted to specific scene categories and cannot synthesize photo-realistic images. Other methods bypass explicit reconstruction, instead focusing on relighting [58] or view synthesis sub-problems [31, 56].

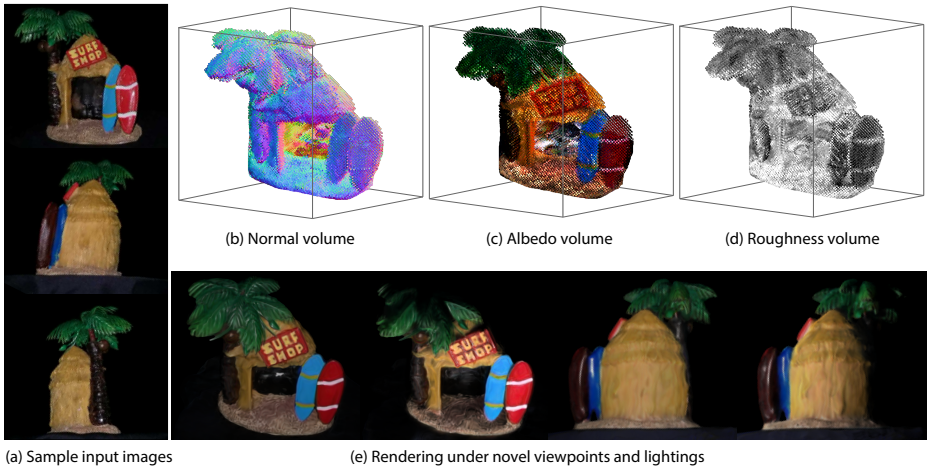

Fig. 1. Given a set of images taken using a mobile phone with flashlight (sampled images are shown in (a)), our method learns a volume representation of the captured object by estimating the opacity volume, normal volume (b) and reflectance volumes such as albedo (c) and roughness (d). Our volume representation enables free navigation of the object under arbitrary viewpoints and novel lighting conditions (e).

Our goal is to make high-quality scene acquisition and rendering practical with off-the-shelf devices under mildly controlled conditions. We use a set of unstructured images captured around a scene by a single mobile phone camera with flash illumination in a dark room. This practical setup acquires multi-view images under collocated viewing and lighting directions—referred to as photometric images [56]. While the high-frequency appearance variation in these images (due to sharp specular highlights and shadows) can result in low-quality mesh reconstruction from state-of-the-art methods (see Fig. 3), we show that our method can accurately model the scene and realistically reproduce complex appearance information like specularities and occlusions.

At the heart of our method is a novel, physically-based neural volume rendering framework. We train a deep neural network that simultaneously learns the geometry and *reflectance* of a scene as volumes. We leverage a decoder-like network architecture, where an encoding vector together with the corresponding network parameters are learned during a per-scene optimization (training) process. Our network decodes a volumetric scene representation consisting of opacity, normal, diffuse color and roughness volumes, which model the global geometry, local surface orientations and spatially-varying reflectance parameters of the scene, respectively. These volumes are supplied to a differentiable rendering module to render images with collocated light-view settings at training time, and arbitrary light-view settings at inference time (see Fig. 2).

We base our differentiable rendering module on classical volume ray marching approaches with opacity (alpha) accumulation and compositing [24, 52]. In

particular, we compute point-wise shading using local normal and reflectance properties, and accumulate the shaded colors with opacities along each marching ray of sight. Unlike the opacity used in previous view synthesis work [31, 62] that is only accumulated along view directions, we propose to learn global scene opacity that can be accumulated from both view and light directions. As shown in Fig. 1, we demonstrate that our scene opacity can be effectively learned and used to compute accurate hard shadows under *novel lighting*, despite the fact that the training process never observed images with shadows that are taken under non-collocated view-light setups. Moreover, different from previous volume-based works [31, 62] that learn a single color at each voxel, we reconstruct per-voxel reflectance and handle complex materials with high glossiness. Our neural rendering framework thus enables rendering with complex view-dependent and light-dependent shading effects including specularities, occlusions and shadows. We compare against a state-of-the-art mesh-based method [37], and demonstrate that our method is able to achieve more accurate reconstructions and renderings (see Fig. 3). We also show that our approach supports scene material editing by modifying the reconstructed reflectance volumes (see Fig. 8). To summarize, our contributions are:

- A practical neural rendering framework that reproduces high-quality geometry and appearance from unstructured mobile phone flash images and enables view synthesis, relighting, and scene editing.
- A novel scene appearance representation using opacity, normal and reflectance volumes.
- A physically-based differentiable volume rendering approach based on deep priors that can effectively reconstruct the volumes from input flash images.

2 Related Works

Geometry reconstruction. There is a long history in reconstructing 3D geometry from images using traditional structure from motion and multi-view stereo (MVS) pipelines [13, 25, 47]. Recently deep learning techniques have also been applied to 3D reconstruction with various representations, including volumes [18, 45], point clouds [1, 42, 51], depth maps [16, 59] and implicit functions [10, 35, 40]. We aim to model scene geometry for realistic image synthesis, for which mesh-based reconstruction [23, 32, 38] is the most common way in many applications [6, 37, 44, 61]. However, it remains challenging to reconstruct accurate meshes for challenging scenes where there are textureless regions and thin structures, and it is hard to incorporate a mesh into a deep learning framework [26, 30]; the few mesh-based deep learning works [15, 20] are limited to category-specific reconstruction and cannot produce photo-realistic results. Instead, we leverage a physically-based opacity volume representation that can be easily embedded in a deep learning system to express scene geometry of arbitrary shapes.

Reflectance acquisition. Reflectance of real materials is classically measured using sophisticated devices to densely acquire light-view samples [12, 33], which

is impractical for common users. Recent works have improved the practicality with fewer samples [39, 57] and more practical devices (mobile phones) [2, 3, 17, 28]; however, most of them focus on flat planar objects. A few single-view techniques based on photometric stereo [4, 14] or deep learning [29] are able to handle arbitrary shape, but they merely recover limited single-view scene content. To recover complete shape with spatially varying BRDF from multi-view inputs, previous works usually rely on a pre-reconstructed initial mesh and images captured under complex controlled setups to reconstruct per-vertex BRDFs [7, 21, 53, 55, 63]. While a recent work [37] uses a mobile phone for practical acquisition like ours, it still requires MVS-based mesh reconstruction, which is ineffective for challenging scenes with textureless, specular and thin-structure regions. In contrast, we reconstruct spatially varying volumetric reflectance via deep network based optimization; we avoid using any initial geometry and propose to jointly reconstruct geometry and reflectance in a holistic framework.

Relighting and view synthesis. Image-based techniques have been extensively explored in graphics and vision to synthesize images under novel lighting and viewpoint without explicit complete reconstruction [8, 11, 27, 43]. Recently, deep learning has been applied to view synthesis and most methods leverage either view-dependent volumes [49, 56, 62] or canonical world-space volumes [31, 48] for geometric-aware appearance inference. We extend them to a more general physically-based volumetric representation which explicitly expresses both geometry and reflectance, and enables relighting with view synthesis. On the other hand, learning-based relighting techniques have also been developed. Purely image-based methods are able to relight scenes with realistic specularities and soft shadows from sparse inputs, but unable to reproduce accurate hard shadows [19, 50, 58, 60]; some other methods [9, 44] propose geometry-aware networks and make use of pre-acquired meshes for relighting and view synthesis, and their performance is limited by the mesh reconstruction quality. A work [36] concurrent to ours models scene geometry and appearance by reconstructing a continuous radiance field for pure view synthesis. In contrast, Deep Reflectance Volumes explicitly express scene geometry and reflectance, and reproduce accurate high-frequency specularities and hard shadows. Ours is the first comprehensive neural rendering framework that enables both relighting and view synthesis with complex shading effects.

3 Rendering with Deep Reflectance Volumes

Unlike a mesh that is comprised of points with complex connectivity, a volume is a regular 3D grid, suitable for convolutional operations. Volumes have been widely used in deep learning frameworks for 3D applications [54, 59]. However, previous neural volumetric representations have only represented pixel colors; this can be used for view synthesis [31, 62], but does not support relighting or scene editing. Instead, we propose to jointly learn geometry and reflectance (i.e. material parameters) volumes to enable broader rendering applications including view synthesis, relighting and material editing in a comprehensive framework.

Deep Reflectance Volumes are learned from a deep network and used to render images in a fully differentiable end-to-end process as shown in Fig. 2. This is made possible by a new differentiable volume ray marching module, which is motivated by physically-based volume rendering. In this section, we introduce our volume rendering method and volumetric scene representation. We discuss how we learn these volumes from unstructured images in Sec. 4.

3.1 Volume rendering overview

In general, volume rendering is governed by the physically-based volume rendering equation (radiative transfer equation) that describes the radiance that arrives at a camera [34, 41]:

$$L(\boldsymbol{c}, \boldsymbol{\omega}_o) = \int_0^\infty \tau(\boldsymbol{c}, \boldsymbol{x})[L_e(\boldsymbol{x}, \boldsymbol{\omega}_o) + L_s(\boldsymbol{x}, \boldsymbol{\omega}_o)]dx, \tag{1}$$

This equation integrates emitted, L_e, and in-scattered, L_s, light contributions along the ray starting at camera position $\boldsymbol{c}$ in the direction $-\boldsymbol{\omega}_o$. Here, x represents distance along the ray, and $\boldsymbol{x} = \boldsymbol{c} - x\boldsymbol{\omega}_o$ is the corresponding 3D point. $\tau(\boldsymbol{c}, \boldsymbol{x})$ is the transmittance factor that governs the loss of light along the line segment between $\boldsymbol{c}$ and $\boldsymbol{x}$:

$$\tau(\boldsymbol{c}, \boldsymbol{x}) = e^{-\int_0^x \sigma_t(z)dz}, \tag{2}$$

where $\sigma_t(z)$ is the extinction coefficient at location z on the segment. The in-scattered contribution is defined as:

$$L_s(\boldsymbol{x}, \boldsymbol{\omega}_o) = \int_{\mathcal{S}} f_p(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \tag{3}$$

in which $\mathcal{S}$ is a unit sphere, $f_p(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ is a generalized (unnormalized) phase function that expresses how light scatters at a point in the volume, and $L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)$ is the incoming radiance that arrives at $\boldsymbol{x}$ from direction $\boldsymbol{\omega}_i$.

In theory, fully computing $L(\boldsymbol{c}, \boldsymbol{\omega}_o)$ requires multiple-scattering computation using Monte Carlo methods [41], which is computationally expensive and unsuitable for deep learning techniques. We consider a simplified case with a single point light, single scattering and no volumetric emission. The transmittance between the scattering location and the point light is handled the same way as between the scattering location and camera. The generalized phase function $f_p(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ becomes a reflectance function $f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{n}(\boldsymbol{x}), R(\boldsymbol{x}))$ which computes reflected radiance at $\boldsymbol{x}$ using its local surface normal $\boldsymbol{n}(\boldsymbol{x})$ and the reflectance parameters $R(\boldsymbol{x})$ of a given surface reflectance model. Therefore, Eqn. 1 and Eqn. 3 can be simplified and written concisely as [24, 34]:

$$L(\boldsymbol{c}, \boldsymbol{\omega}_o) = \int_0^\infty \tau(\boldsymbol{c}, \boldsymbol{x})\tau(\boldsymbol{x}, \boldsymbol{l}) f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{n}(\boldsymbol{x}), R(\boldsymbol{x})) L_{\boldsymbol{l}}(\boldsymbol{x}, \boldsymbol{\omega}_i) dx, \tag{4}$$

where $\boldsymbol{l}$ is the light position, $\boldsymbol{\omega}_i$ corresponds to the direction from $\boldsymbol{x}$ to $\boldsymbol{l}$, $\tau(\boldsymbol{c}, \boldsymbol{x})$ still represents the transmittance from the scattering point $\boldsymbol{x}$ to the camera $\boldsymbol{c}$, the

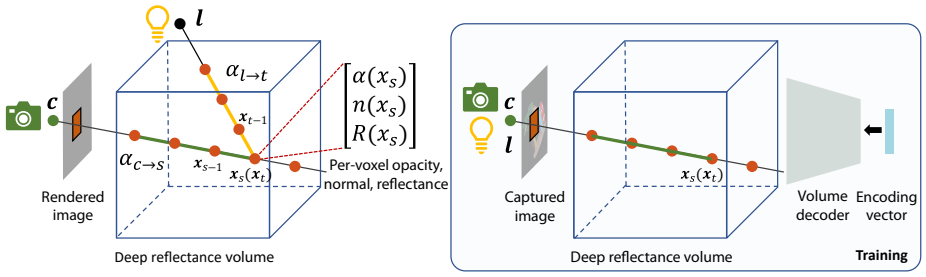

Fig. 2. We propose Deep Reflectance Volume representation to capture scene geometry and appearance, where each voxel consists of opacity α, normal n and reflectance (material coefficients) R. During rendering, we perform ray marching through each pixel and accumulate contributions from each point $\boldsymbol{x}_s$ along the ray. Each contribution is calculated using the local normal, reflectance and lighting information. We accumulate opacity from both the camera $\alpha_{c\to s}$ and the light $\alpha_{l\to t}$ to model the light transport loss in both occlusions and shadows. To predict such a volume, we start from an encoding vector, and decode it into a volume using a 3D convolutional neural network; thus the combination of the encoding vector and network weights is the unknown variable being optimized (trained). We train on images captured with collocated camera and light by enforcing a loss function between rendered images and training images.

term $\tau(\boldsymbol{x}, \boldsymbol{l})$ (that was implicitly involved in Eqn. 3) is the transmittance from the light $\boldsymbol{l}$ to $\boldsymbol{x}$ and expresses light extinction before scattering, and $L_{\boldsymbol{l}}(\boldsymbol{x}, \boldsymbol{\omega}_i)$ represents the light intensity arriving at $\boldsymbol{x}$ without considering light extinction.

3.2 A discretized, differentiable volume rendering module

To make volume rendering practical in a learning framework, we further approximate Eqn. 4 by turning it into a discretized version, which can be evaluated by ray marching [24, 34, 52]. This is classically expressed using opacity compositing, where opacity α is used to represent the transmittance with fixed ray marching step size Δx. Points are sequentially sampled along a given ray, $\boldsymbol{\omega}_o$ from the camera position, $\boldsymbol{c}$ as:

$$\boldsymbol{x}_s = \boldsymbol{x}_{s-1} - \boldsymbol{\omega}_o \Delta x = \boldsymbol{c} - s\boldsymbol{\omega}_o \Delta x. \tag{5}$$

The radiance L_s and opacity $\alpha_{c\to s}$ along this path, $c \to s$, are recursively accumulated until $\boldsymbol{x}_s$ exits the volume as:

$$L_s = L_{s-1} + [1 - \alpha_{c\to(s-1)}][1 - \alpha_{l\to(t-1)}]\alpha(\boldsymbol{x}_s)L(\boldsymbol{x}_s), \tag{6}$$

$$\alpha_{c\to s} = \alpha_{c\to(s-1)} + [1 - \alpha_{c\to(s-1)}]\alpha(\boldsymbol{x}_s), \tag{7}$$

$$L(\boldsymbol{x}_s) = f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{n}(\boldsymbol{x}_s), R(\boldsymbol{x}_s))L_{\boldsymbol{l}}(\boldsymbol{x}_s, \boldsymbol{\omega}_i). \tag{8}$$

Here, $L(\boldsymbol{x}_s)$ computes the reflected radiance from the reflectance function and the incoming light, $\alpha_{c\to s}$ represents the accumulated opacity from the camera $\boldsymbol{c}$

to point $\boldsymbol{x}_s$, and corresponds to $\tau(\boldsymbol{c}, \boldsymbol{x})$ in Eqn 4. $\alpha_{l \to t}$ represents the accumulated opacity from the light $\boldsymbol{l}$—i.e., $\tau(\boldsymbol{x}, \boldsymbol{l})$ in Eqn. 4—and requires a *separate* accumulation process over samples along the $\boldsymbol{l} \to \boldsymbol{x}_s$ ray, similar to Eqn. 7:

$$\boldsymbol{x}_s = \boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \boldsymbol{\omega}_i \Delta x = \boldsymbol{l} - t\boldsymbol{\omega}_i \Delta x, \tag{9}$$

$$\alpha_{l \to t} = \alpha_{l \to (t-1)} + [1 - \alpha_{l \to (t-1)}]\alpha(\boldsymbol{x}_t). \tag{10}$$

In this rendering process (Eqn. 5-10), a scene is represented by an opacity volume α, a normal volume $\boldsymbol{n}$ and a BRDF volume R; together, these express the geometry and reflectance of the scene, and we refer to them as *Deep Reflectance Volumes*. The simplified opacity volume α is essentially one minus the transmission τ (depending on the physical extinction coefficient σ_t) over a ray segment of a fixed step size Δx; this means that α is dependent on Δx.

Our physically-based ray marching is fully differentiable, so it can be easily incorporated in a deep learning framework and backpropagated through. With this rendering module, we present a neural rendering framework that simultaneously learns scene geometry and reflectance from captured images.

We support any differentiable reflectance model f_r and, in practice, use the simplified Disney BRDF model [22] that is parameterized by diffuse albedo and specular roughness (please refer to the supplementary materials for more details). Our opacity volume is a general geometry representation, accounting for both occlusions (view opacity accumulation in Eqn. 7) and shadows (light opacity accumulation in Eqn. 10). We illustrate our neural rendering with ray marching in Fig. 2. Note that, because our acquisition setup has collocated camera and lighting, $\alpha_{l \to t}$ becomes equivalent to $\alpha_{c \to s}$ during training, thus requiring only one-pass opacity accumulation from the camera. However, the learned opacity can still be used for re-rendering under any *non-collocated lighting* with two-pass opacity accumulation.

Note that while alpha compositing-based rendering functions have been used in previous work on view synthesis, their formulations are not physically-based [31] and are simplified versions that don't model lighting [49, 62]. In contrast, our framework is physically-based and models single-bounce light transport with complex reflectance, occlusions and shadows.

4 Learning Deep Reflectance Volumes

4.1 Overview

Given a set of images of a real scene captured under multiple known viewpoints with collocated lighting, we propose to use a neural network to reconstruct a Deep Reflectance Volume representation of a real scene. Similar to Lombardi et al. [31], our network starts from a 512-channel deep encoding vector that encodes scene appearance; in contrast to their work, where this volume only represents RGB colors, we decode a vector to an opacity volume α, normal volume $\boldsymbol{n}$ and reflectance volume R for rendering. Moreover, our scene encoding vector is

not predicted by any network encoder; instead, we jointly optimize for a scene encoding vector and scene-dependent decoder network.

Our network infers the geometry and reflectance volumes in a transformed 3D space with a learned warping function W. During training, our network learns the warping function W, and the geometry and reflectance volumes α_w, $\boldsymbol{n}_w$, R_w, where the subscript w refers to a volume in the warped space. The corresponding world-space scene representation is expressed by $V(\boldsymbol{x}) = V_w(W(\boldsymbol{x}))$, where V is α, $\boldsymbol{n}$ or R. In particular, we use bilinear interpolation to fetch a corresponding value at an arbitrary position $\boldsymbol{x}$ in the space from the discrete voxel values. We propose a decoder-like network, which learns to decode the warping function and the volumes from the deep scene encoding vector. We use a rendering loss between rendered and captured images as well as two regularizing terms.

4.2 Network architecture

Geometry and reflectance. To decode the geometry and reflectance volumes (α_w, $\boldsymbol{n}_w$, R_w), we use upsampling 3D convolutional operations to 3D-upsample the deep scene encoding vector to a multi-channel volume that contains the opacity, normal and reflectance. In particular, we use multiple transposed convolutional layers with stride 2 to upsample the volume, each of which is followed by a LeakyRelu activation layer. The network regresses an 8-channel $128 \times 128 \times 128$ volume that includes α_w, $\boldsymbol{n}_w$ and R_w—one channel for opacity α_w, three channels for normal $\boldsymbol{n}_w$, and four channels for reflectance R_w (three for albedo and one for roughness). These volumes express the scene geometry and reflectance in a transformed space, which can be warped to the world space for ray marching.

Warping function. To increase the effective resolution of the volume, we learn an affine-based warping function similar to [31]. The warping comprises a global warping and a spatially-varying warping. The global warping is represented by an affine transformation matrix W_g. The spatially varying warping is modeled in the inverse transformation space, which is represented by six basis affine matrices $\{W_j\}_{j=1}^{16}$ and a $32 \times 32 \times 32$ 16-channel volume B that contains spatially-varying linear weights of the 16 basis matrices. Specifically, given a world-space position $\boldsymbol{x}$, the complete warping function W maps it into a transformed space by:

$$W(\boldsymbol{x}) = [\sum_{j=1}^{16} B_j(\boldsymbol{x}) W_j]^{-1} W_g \boldsymbol{x}, \tag{11}$$

where $B_j(\boldsymbol{x})$ represents the normalized weight of the jth warping basis at $\boldsymbol{x}$. Here, each global or local basis affine transformation matrix W_* is composed of rotation, translation and scale parameters, which are optimized during the training process. Our network decodes the weight volume B from the deep encoding vector using a multi-layer perceptron network with fully connected layers.

4.3 Loss function and training details

Loss function. Our network learns the scene volumes using a rendering loss computed using the differentiable ray marching process discussed in Sec. 3. During training, we randomly sample pixels from the captured images and do the ray marching (using known camera calibration) to get the rendered pixel colors L_k of pixel k; we supervise them with the ground truth colors $\tilde{L}_k$ in the captured images using a L_2 loss. In addition, we also apply regularization terms from additional priors similar to [31]. We only consider opaque objects in this work and enforce the accumulated opacity along any camera ray $\alpha_{c_k \to s'}$ (see Eqn. 7, here k denotes a pixel and s' reflects the final step that exits the volume) to be either 0 or 1, corresponding to a background or foreground pixel, respectively. We also regularize the per-voxel opacity to be sparse over the space by minimizing the spatial gradients of the logarithmic opacity. Our total loss function is given by:

$$\sum_k \|L_k - \tilde{L}_k\|^2 + \beta_1 \sum_k [\log(\alpha_{c_k \to s'}) + \log(1 - \alpha_{c_k \to s'})] + \beta_2 \sum \|\nabla_{\boldsymbol{x}} \log \alpha(\boldsymbol{x})\| \quad (12)$$

Here, the first part reflects the data term, the second regularizes the accumulated α and the third regularizes the spatial sparsity.

Training details. We build our volume as a cube located at $[-1, 1]^3$. During training, we randomly sample 128×128 pixels from 8 captured images for each training batch, and perform ray marching through the volume using a step size of $1/64$. Initially, we set $\beta_1 = \beta_2 = 0.01$; we increase these weights to $\beta_1 = 1.0$, $\beta_2 = 0.1$ after 300000 iterations, which helps remove the artifacts in the background and recover sharp boundaries.

5 Results

In this section we show our results on real captured scenes. We first introduce our acquisition setup and data pre-processing. Then we compare against the state-of-the-art mesh-based appearance acquisition method, followed by a detailed analysis of the experiments. We also demonstrate material editing results with our approach. Please refer to the supplementary materials for video results.

Data acquisition. Our approach learns the volume representation in a scene dependent way from images with collocated view and light; this requires adequately dense input images well distributed around a target scene to learn complete appearance. Such data can be practically acquired by shooting a video using a handheld cellphone; we show one result using this practical handheld setup in Fig. 4. For other results, we use a robotic arm to automatically capture more uniformly distributed images around scenes for convenience and thorough evaluations; this allows us to evaluate the performance of our method with different numbers of input images that are roughly uniformly distributed as shown in Tab. 5. In the robotic arm setups, we mount a Samsung Galaxy Note 8 cellphone to the robotic arm and capture about 480 images using its camera and

the built-in flashlight in a dark room; we leave out a subset of 100 images for validation purposes and use the others for training. We use the same phone to capture a 4-minute video of the object in CAPTAIN and select one image for training for every 20 frames, which effectively gives us 310 training images.

Data pre-processing. Our captured objects are roughly located around the center of the images. We select one fixed rectangular region around the center that covers the object across all frames and use it to crop the images as input for training. The resolution of the cropped training images fed to our network ranges from 400×500 to 1100×1100. Note that we do not use a foreground mask for the object. Our method leverages the regularization terms in training (see Sec. 4.3), which automatically recovers a clean background. We calibrate the captured images using structure from motion (SfM) in COLMAP [46] to get the camera intrinsic and extrinsic parameters. Since SfM may fail to register certain views, the actual number of training images varies from 300 to 385 in different scenes. We estimate the center and bounding box of the captured object with the sparse reconstructions from SfM. We translate the center of the object to the origin and scale it to fit into the $[-1, 1]^3$ cube.

Implementation and timing. We implement our system (both neural network and differentiable volume rendering components) using PyTorch. We train our network using four NVIDIA 2080Ti RTX GPUs for about two days (about 450000 iterations; though 200000 iterations for 1 day typically already converges to good results, see Fig. 7). At inference time, we directly render the scene from the reconstructed volumes without the network. It takes about 0.8s to render a 700×700 image under collocated view and light. For non-collocated view and light, the rendering requires connecting each shading point to the light source with additional light-dependent opacity accumulation, which is very expensive if done naively. To facilitate this process, we perform ray marching from the light's point of view and precompute the accumulated opacity at each spatial position of the volume. During rendering, the accumulated opacity for the light ray can be directly sampled from the precomputed volume. By doing so, our final rendering under arbitrary light and view takes about 2.3s.

Comparisons with mesh-based reconstruction. We use a practical acquisition setup where we capture unstructured images using a mobile phone with its built-in flashlight on in a dark room. Such a mildly controlled acquisition setup is rarely supported by previous works [7, 21, 55, 56, 58, 63]. Therefore, we compare with the state-of-the-art method proposed by Nam et al. [37] for mesh-based geometry and reflectance reconstruction, that uses the same cellphone setup as ours to reconstruct a mesh with per-vertex BRDFs, and supports both relighting and view synthesis. Figure 3 shows comparisons on renderings under both collocated and non-collocated view-light conditions. The comparison results are generated from the same set of input images, and we requested the authors of [37] run their code on our data and compared on the rendered images provided by the authors. Please refer to the supplementary materials for video comparisons.

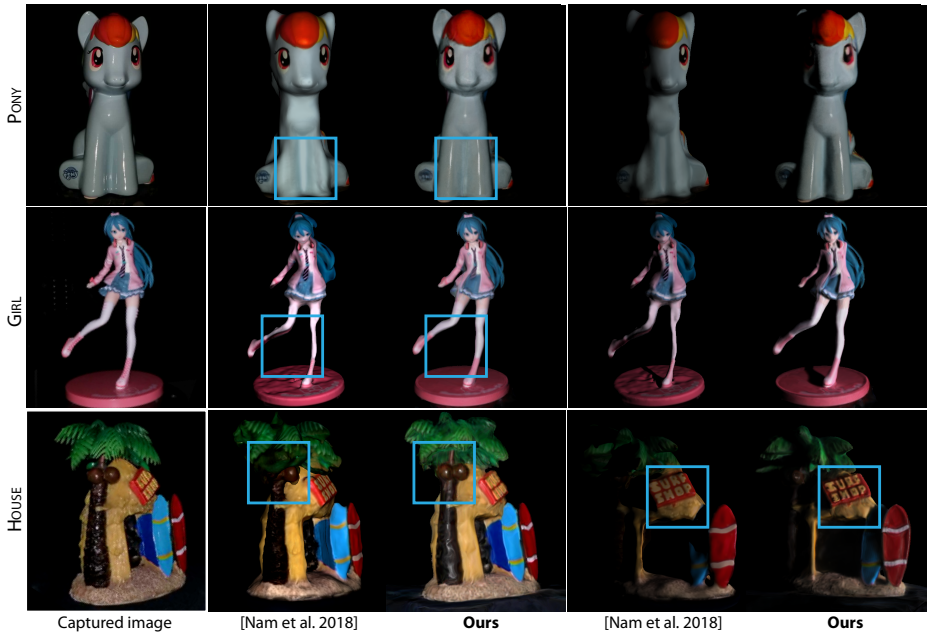

Fig. 3. Comparisons with mesh-based reconstruction. We show renderings of the captured object under both collocated (column 2, 3) and non-collocated (column 4, 5) camera and light. We compare our volume-based neural reconstruction against a state-of-the-art method [37] that reconstructs mesh and per-vertex BRDFs. Nam et al. [37] fails to handle such challenging cases and recovers inaccurate geometry and appearance. In contrast our method produces photo-realistic results.

As shown in Fig. 3, our results are significantly better than the mesh-based method in terms of both geometry and reflectance. Note that, Nam et al. [37] leverage a state-of-the-art MVS method [47] to reconstruct the initial mesh from captured images and performs an optimization to further refine the geometry; this however still fails to recover the accurate geometry in texture-less, specular and thin-structured regions in those challenging scenes, which leads to seriously distorted shapes in Pony, over-smoothness and undesired structures in House, and degraded geometry in Girl. Our learning-based volumetric representation avoids these mesh-based issues and models the scene geometry accurately with many details. Moreover, it is also very difficult for the classical per-vertex BRDF optimization in [37] to recover high-frequency specularities, which leads to over-diffuse appearance in most of the scenes; this is caused by the lack of constraints for the high-frequency specular effects, which appear in very few pixels in limited input views. In contrast, our optimization is driven by our novel neural rendering framework with deep network priors, which effectively correlates the sparse specularities in different regions through network connections and recovers realistic specularities and other appearance effects.

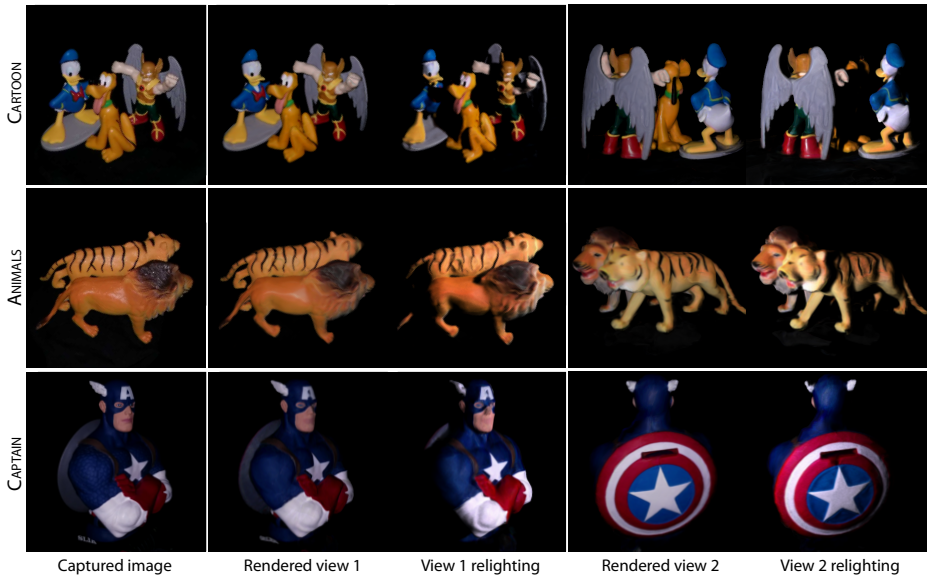

Fig. 4. Additional results on real scenes. We show renderings under novel view and lighting conditions. Our method is able to handle scenes with multiple objects (top two rows) and model the complex occlusions between them. Our method can also generate high-quality results from casual handheld video captures (third row), which demonstrates the practicability of our approach.

	25	50	100	200	385
PSNR	25.33	26.36	26.95	27.85	28.13
SSIM	0.70	0.73	0.75	0.80	0.81

Fig. 5. We evaluate the performance of our method on the House scene with different numbers of training images. Although we use all 385 images in our final experiments, our method is able to achieve comparable performance with as few as 200 images for this challenging scene.

	House	Cartoon
[48]	0.786/25.81	0.532/16.34
Ours	**0.896/30.44**	**0.911/29.14**

Fig. 6. We compare against DeepVoxels on synthesizing novel views under collocated lights and report the PSNR/SSIM scores. The results show that our method generates more accurate renderings. Note that we retrain our model with a resolution of 512×512 for a fair comparison.

Comparison on synthesizing novel views. We also make a comparison on synthesizing novel views under collocated lights against a view synthesis method DeepVoxels [48], which encodes view-dependent appearance in a learnt 3D-aware neural representation. Note that DeepVoxels does not support relighting. As shown in Fig. 6, our method is able to generate renderings of higher quality with higher PSNR/SSIM scores. In contrast, DeepVoxels fails to reason about the complex geometry in our real scenes, thus resulting in degraded image quality. Please refer to the supplementary materials for visual comparison results.

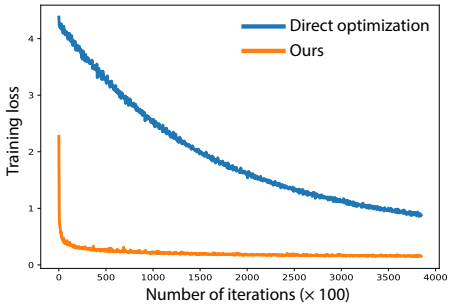

Fig. 7. We compare our deep prior based optimization against direct optimization of the volume and warping function without using networks. Direct optimization converges significantly slower than our method, which demonstrates the effectiveness of regularization by the networks.

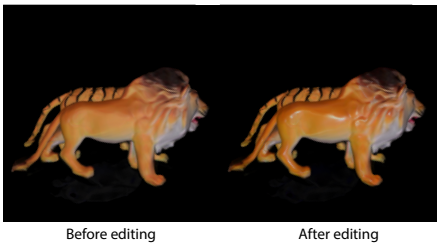

Fig. 8. Our approach supports intuitive editing of the material properties of a captured object. In this example we decrease the roughness of the object to make it look like glossy marble instead of plastic.

Additional results. We show additional relighting and view synthesis results of complex real scenes in Fig. 4. Our method is able to handle scenes with multiple objects, as shown in scene CARTOON and ANIMALS. Our volumetric representation can accurately model complex occlusions between objects and reproduce realistic cast shadows under novel lighting, which are never observed by our network during the training process. In the CAPTAIN scene, we show the result generated from *handheld* mobile phone captures. We select frames from the video at fixed intervals as training data. Despite the potential existence of motion blur and non-uniform coverage, our method is able to generate high-quality results, which demonstrates the robustness and practicality of our approach. Please refer to the supplementary materials for video results.

Evaluation of the number of inputs. Our method relies on an optimization over adequate input images that capture the scene appearance across different view/light directions. We evaluate how our reconstruction degrades with the decrease of training images on the HOUSE scene. We uniformly select a subset of views from the full training images and train our model on them. We evaluate the trained model on the test images, and report the SSIMs and PSNRs in Fig. 5. As we can see from the results, there is an obvious performance drop when there are fewer than 100 training images due to insufficient constraints. On the other hand, while we use the full 385 images for our final results, our method in fact achieves comparable performance with only 200 for this scene, as reflected by their close PSNRs and SSIMs.

Comparison with direct optimization. Our neural rendering leverages a "deep volume prior" to drive the volumetric optimization process. To justify the effectiveness of this design, we compare with a naive method that directly opti-

mizes the parameters in each voxel and the warping parameters using the same loss function. We show the optimization progress in Fig. 7. Note that, the naive method converges significantly slower than ours, where the independent voxel-wise optimization without considering across-voxel correlations cannot properly disentangle the ambiguous information in the captured images; yet, our deep optimization is able to correlate appearance information across the voxels with deep convolutions, which effectively minimizes the reconstruction loss.

Material editing. Our method learns explicit volumes with physical meaning to represent the reflectance of real scenes. This enables broad image synthesis applications like editing the materials of captured scenes. We show one example in Fig. 8, where we successfully make the scene glossier by decreasing the learned roughness in the volume. Note that, the geometry and colors are still preserved in the scene, while novel specularities are introduced which are not part of the material appearance in the scene. This example illustrates that our network disentangles the geometry and reflectance of the scene in a reasonable way, thereby enabling sub-scene component editing without influencing other components.

Limitations. We reconstruct the deep reflectance volumes with a resolution of 128^3, which is restricted by available GPU memory. While we have applied a warping function to increase the actual utilization of the volume space, and demonstrated that it is able to generate compelling results on complex real scenes, it may fail to fully reproduce the geometry and appearance of scenes with highly complex surface normal variations and texture details. Increasing the volume resolution may resolve this issue. In the future, it would also be interesting to investigate how to efficiently apply sparse representations such as octrees in our framework to increase the capacity of our volume representation. The current reflectance model we are using is most appropriate for opaque surfaces. Extensions to other materials like hair, fur or glass could be potentially addressed by applying other reflectance models in our neural rendering framework.

6 Conclusion

We have presented a novel approach to learn a volume representation that models both geometry and reflectance of complex real scenes. We predict per-voxel opacity, normal, and reflectance from unstructured multi-view mobile phone captures with the flashlight. We also introduce a physically-based differentiable rendering module to enable renderings of the volume under arbitrary viewing and lighting directions. Our method is practical, and supports novel view synthesis, relighting and material editing, which has significant potential benefits in scenarios such as 3D visualization and VR/AR applications.

Acknowledgements. We thank Giljoo Nam for help with the comparisons. This work was supported in part by ONR grants N000141712687, N000141912293, N000142012529, NSF grant 1617234, Adobe, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: ICML. pp. 40–49 (2018)
2. Aittala, M., Aila, T., Lehtinen, J.: Reflectance modeling by neural texture synthesis. ACM Transaction on Graphics **35**(4), 65:1–65:13 (Jul 2016)
3. Aittala, M., Weyrich, T., Lehtinen, J.: Two-shot svbrdf capture for stationary materials. ACM Transactions on Graphics **34**(4), 110:1–110:13 (Jul 2015)
4. Alldrin, N., Zickler, T., Kriegman, D.: Photometric stereo with non-parametric and spatially-varying reflectance. In: CVPR. pp. 1–8. IEEE (2008)
5. Baek, S.H., Jeon, D.S., Tong, X., Kim, M.H.: Simultaneous acquisition of polarimetric SVBRDF and normals. ACM Transactions on Graphics **37**(6), 268–1 (2018)
6. Bi, S., Kalantari, N.K., Ramamoorthi, R.: Patch-based optimization for image-based texture mapping. ACM Transaction on Graphics **36**(4), 106–1 (2017)
7. Bi, S., Xu, Z., Sunkavalli, K., Kriegman, D., Ramamoorthi, R.: Deep 3d capture: Geometry and reflectance from sparse multi-view images. In: CVPR. pp. 5960–5969 (2020)
8. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: SIGGRAPH. pp. 425–432. ACM (2001)
9. Chen, Z., Chen, A., Zhang, G., Wang, C., Ji, Y., Kutulakos, K.N., Yu, J.: A neural rendering framework for free-viewpoint relighting. In: CVPR (June 2020)
10. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. arXiv preprint arXiv:1812.02822 (2018)
11. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 145–156. ACM Press/Addison-Wesley Publishing Co. (2000)
12. Foo, S.C.: A gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation. Ph.D. thesis, Citeseer (1997)
13. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE transactions on pattern analysis and machine intelligence **32**(8), 1362–1376 (2009)
14. Goldman, D.B., Curless, B., Hertzmann, A., Seitz, S.M.: Shape and spatially-varying brdfs from photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(6), 1060–1071 (2009)
15. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3D surface generation. In: CVPR. pp. 216–224 (2018)
16. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: DeepMVS: Learning multi-view stereopsis. In: CVPR. pp. 2821–2830 (2018)
17. Hui, Z., Sunkavalli, K., Lee, J.Y., Hadap, S., Wang, J., Sankaranarayanan, A.C.: Reflectance capture using univariate sampling of brdfs. In: ICCV. pp. 5362–5370 (2017)
18. Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In: ICCV. pp. 2307–2315 (2017)
19. Kanamori, Y., Endo, Y.: Relighting humans: occlusion-aware inverse rendering for full-body human images. ACM Transactions on Graphics **37**(6), 1–11 (2018)
20. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 371–386 (2018)
21. Kang, K., Xie, C., He, C., Yi, M., Gu, M., Chen, Z., Zhou, K., Wu, H.: Learning efficient illumination multiplexing for joint capture of reflectance and shape. (2019)

22. Karis, B., Games, E.: Real shading in unreal engine 4
23. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
24. Kniss, J., Premoze, S., Hansen, C., Shirley, P., McPherson, A.: A model for volume lighting and modeling. IEEE transactions on visualization and computer graphics **9**(2), 150–162 (2003)
25. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. ICCV **38**(3), 199–218 (2000)
26. Ladicky, L., Saurer, O., Jeong, S., Maninchedda, F., Pollefeys, M.: From point clouds to mesh using regression. In: ICCV. pp. 3893–3902 (2017)
27. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 31–42. ACM (1996)
28. Li, Z., Sunkavalli, K., Chandraker, M.: Materials for masses: SVBRDF acquisition with a single mobile phone image. In: ECCV. pp. 72–87 (2018)
29. Li, Z., Xu, Z., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. In: SIGGRAPH Asia 2018. p. 269. ACM (2018)
30. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: CVPR. pp. 2916–2925 (2018)
31. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM Transactions on Graphics **38**(4), 65 (2019)
32. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
33. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. ACM Transactions on Graphics **22**(3), 759–769 (Jul 2003)
34. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)
35. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. arXiv preprint arXiv:1812.03828 (2018)
36. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis (2020)
37. Nam, G., Lee, J.H., Gutierrez, D., Kim, M.H.: Practical SVBRDF acquisition of 3D objects with unstructured flash photography. In: SIGGRAPH Asia 2018. p. 267. ACM (2018)
38. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality. pp. 127–136. ISMAR '11, IEEE Computer Society, Washington, DC, USA (2011)
39. Nielsen, J.B., Jensen, H.W., Ramamoorthi, R.: On optimal, minimal brdf sampling for reflectance acquisition. ACM Transactions on Graphics **34**(6), 1–11 (2015)
40. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR. pp. 3504–3515 (2020)
41. Novák, J., Georgiev, I., Hanika, J., Jarosz, W.: Monte carlo methods for volumetric light transport simulation. In: Computer Graphics Forum. vol. 37, pp. 551–576. Wiley Online Library (2018)

42. Paschalidou, D., Ulusoy, O., Schmitt, C., Van Gool, L., Geiger, A.: Raynet: Learning volumetric 3d reconstruction with ray potentials. In: CVPR. pp. 3897–3906 (2018)
43. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. ACM Transactions on Graphics **28**(1), 3 (2009)
44. Philip, J., Gharbi, M., Zhou, T., Efros, A.A., Drettakis, G.: Multi-view relighting using a geometry-aware network. ACM Transactions on Graphics **38**(4), 1–14 (2019)
45. Richter, S.R., Roth, S.: Matryoshka networks: Predicting 3D geometry via nested shape layers. In: CVPR. pp. 1936–1944 (2018)
46. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
47. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
48. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3D feature embeddings. In: CVPR. pp. 2437–2446 (2019)
49. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: CVPR. pp. 175–184 (2019)
50. Sun, T., Barron, J.T., Tsai, Y.T., Xu, Z., Yu, X., Fyffe, G., Rhemann, C., Busch, J., Debevec, P., Ramamoorthi, R.: Single image portrait relighting. ACM Transactions on Graphics (Proceedings SIGGRAPH) (2019)
51. Wang, J., Sun, B., Lu, Y.: Mvpnet: Multi-view point regression networks for 3D object reconstruction from a single image. arXiv preprint arXiv:1811.09410 (2018)
52. Wittenbrink, C.M., Malzbender, T., Goss, M.E.: Opacity-weighted color interpolation, for volume sampling. In: Proceedings of the 1998 IEEE symposium on Volume visualization. pp. 135–142 (1998)
53. Wu, H., Wang, Z., Zhou, K.: Simultaneous localization and appearance estimation with a consumer rgb-d camera. IEEE Transactions on visualization and computer graphics **22**(8), 2012–2023 (2015)
54. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D shapenets: A deep representation for volumetric shapes. In: CVPR. pp. 1912–1920 (2015)
55. Xia, R., Dong, Y., Peers, P., Tong, X.: Recovering shape and spatially-varying surface reflectance under unknown illumination. ACM Transactions on Graphics **35**(6), 187 (2016)
56. Xu, Z., Bi, S., Sunkavalli, K., Hadap, S., Su, H., Ramamoorthi, R.: Deep view synthesis from sparse photometric images. ACM Transactions on Graphics **38**(4), 76 (2019)
57. Xu, Z., Nielsen, J.B., Yu, J., Jensen, H.W., Ramamoorthi, R.: Minimal brdf sampling for two-shot near-field reflectance acquisition. ACM Transactions on Graphics **35**(6), 188 (2016)
58. Xu, Z., Sunkavalli, K., Hadap, S., Ramamoorthi, R.: Deep image-based relighting from optimal sparse samples. ACM Transactions on Graphics **37**(4), 126 (2018)
59. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: MVSNet: Depth inference for unstructured multi-view stereo. In: ECCV. pp. 767–783 (2018)
60. Zhou, H., Hadap, S., Sunkavalli, K., Jacobs, D.W.: Deep single-image portrait relighting. In: ICCV. pp. 7194–7202 (2019)
61. Zhou, Q.Y., Koltun, V.: Color map optimization for 3D reconstruction with consumer depth cameras. ACM Transactions on Graphics **33**(4), 155 (2014)

62. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: learning view synthesis using multiplane images. ACM Transactions on Graphics **37**(4), 1–12 (2018)
63. Zhou, Z., Chen, G., Dong, Y., Wipf, D., Yu, Y., Snyder, J., Tong, X.: Sparse-as-possible SVBRDF acquisition. ACM Transactions on Graphics **35**(6), 189 (2016)