Deferred Neural Lighting: Free-viewpoint Relighting from Unstructured Photographs

DUAN GAO, BNRist, Tsinghua University and Microsoft Research Asia
GUOJUN CHEN, Microsoft Research Asia
YUE DONG, Microsoft Research Asia
PIETER PEERS, College of William & Mary
KUN XU, BNRist, Tsinghua University
XIN TONG, Microsoft Research Asia

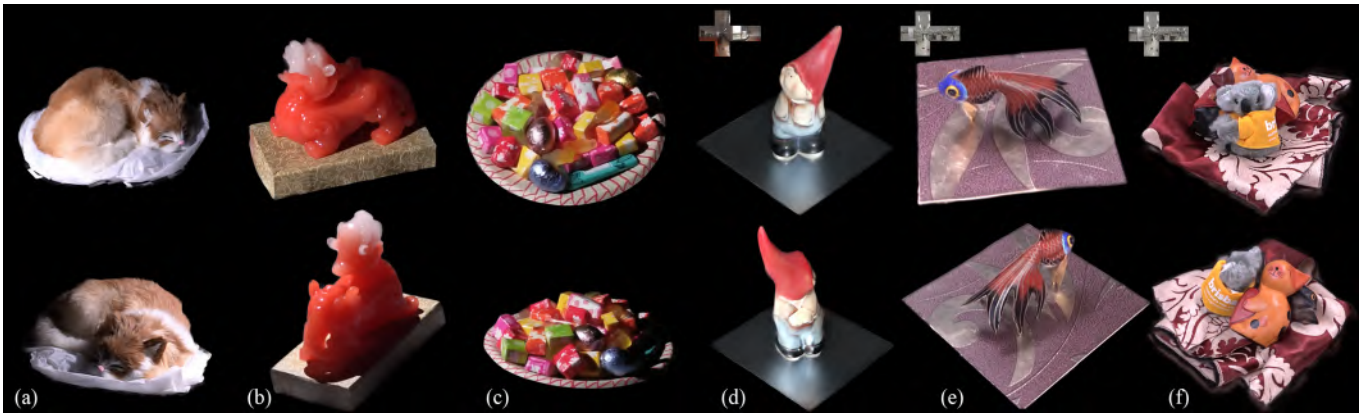

Fig. 1. Examples of scenes captured with a handheld dual camera setup, and relit using deferred neural lighting. (a) A cat with non-polygonal shape (i.e., fur), (b) a translucent Pixiu statuette, (c) and a candy bowl scene with complex shadowing. All three scenes are relit with a directional light. (d) A gnome statue on a glossy surface, (e) a decorative fish on a spatially varying surface, and (f) a cluttered scene with a stuffed Koala toy, a wooden toy cat and anisotropic satin. All three scenes are relit by the natural environment maps shown in the insets.

We present deferred neural lighting, a novel method for free-viewpoint relighting from unstructured photographs of a scene captured with handheld devices. Our method leverages a scene-dependent neural rendering network for relighting a rough geometric proxy with learnable neural textures. Key to making the rendering network lighting aware are radiance cues: global illumination renderings of a rough proxy geometry of the scene for a small set of basis materials and lit by the target lighting. As such, the light transport through the scene is never explicitely modeled, but resolved at rendering time by a neural rendering network. We demonstrate that the neural textures and neural renderer can be trained end-to-end from unstructured photographs captured with a double hand-held camera setup that concurrently captures the scene while being lit by only one of the cameras' flash lights. In addition, we propose a novel augmentation refinement strategy that exploits the linearity of light transport to extend the relighting capabilities of the neural rendering network to support other lighting types (e.g., environment lighting) beyond the lighting used during acquisition (i.e., flash lighting). We demonstrate our deferred neural lighting solution on a variety of real-world and synthetic scenes exhibiting a wide range of material properties, light transport effects, and geometrical complexity.

CCS Concepts: • **Computing methodologies** → **Image-based rendering**; **Reflectance modeling**.

Additional Key Words and Phrases: Relighting, Free-viewpoint, Neural Rendering

ACM Reference Format:
Duan Gao, Guojun Chen, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2020. Deferred Neural Lighting: Free-viewpoint Relighting from Unstructured Photographs. *ACM Trans. Graph.* 39, 6, Article 258 (December 2020), 15 pages. https://doi.org/10.1145/3414685.3417767

1 INTRODUCTION

Digitally reproducing the appearance of a scene from a novel viewpoint and under novel lighting is a challenging research problem with many practical applications in both computer graphics as well

Part of this work was performed while Pieter Peers visited Microsoft Research Asia.
Authors' addresses: Duan Gao, Tsinghua University, Beijing, China, gaod17@mails.tsinghua.edu.cn; Guojun Chen, Microsoft Research Asia, Beijing, China, guoch@microsoft.com; Yue Dong, Microsoft Research Asia, Beijing, China, yuedong@microsoft.com; Pieter Peers, College of William & Mary, Virginia, USA, ppeers@siggraph.org; Kun Xu, Tsinghua University, Beijing, China xukun@tsinghua.edu.cn; Xin Tong, Microsoft Research Asia, Beijing, China, xtong@microsoft.com.

0730-0301/2020/12-ART258 $15.00
https://doi.org/10.1145/3414685.3417767

as in computer vision. The classic model-based approach for digitizing the appearance of a scene is to quantify each component that impacts the scene appearance, i.e., shape, material properties, and lighting [Lensch et al. 2003; Nam et al. 2018; Xia et al. 2016]. However, due to practical or model limitations, it might not be possible to recover each of these components exactly, thereby potentially affecting the revisualization accuracy of the scene. For example, it might not be possible to obtain an unoccluded view of each part of the shape, thereby yielding an incorrect geometry estimate which in turn will affect the global light transport simulation through the scene. In contrast to model-based approaches, image-based solutions directly modify and recast the appearance information contained in measurements of the scene to the target rendering, thereby inheriting all the intricate light transport details present in the measurements [Debevec et al. 2000; Gortler et al. 1996; Levoy and Hanrahan 1996]. However, to accurately reproduce the correct appearance, generally a dense and accurately calibrated sampling of the scene's appearance is needed, which might be costly or even impractical to obtain. Both approaches have benefitted from new advances in neural networks to increase the capabilities and reduce acquisition costs [Gao et al. 2019; Kang et al. 2019; Xu et al. 2018].

In this paper we present a novel image-based method for 360° *free-viewpoint relighting from unstructured photographs* that borrows ideas from model-based approaches, without the stringent accuracy demands on the components, and that leverages neural networks to reduce the complexity of typical image-based acquisition procedures. We take inspiration from *deferred lighting* [Geldreich et al. 2014], a variant of deferred rendering in which an additional lighting pass is performed before generating the final rendered image, and combine it with the concept of neural textures [Thies et al. 2019]. Similar to deferred neural rendering [Thies et al. 2019], we project learned neural textures in the first pass onto a rough proxy geometry. In a second *"lighting pass"*, we compute a small set of rough radiance cues by rendering a set of predetermined (scene-independent) homogeneous basis materials on the rough geometry under the desired target lighting. Finally, we combine the radiance cues with the projected neural textures, and forward them to a scene-dependent learned neural rendering network to produce the final relit results. The goal of the rough proxy geometry and radiance cues are to aid the neural rendering network in view-synthesis and image-based relighting. In particular, the radiance cues play a crucial role in generalizing the neural rendering network with respect to appearance-changes under novel lighting or viewpoint, as they naturally encode this information.

We train our neural representation directly on unstructured photographs of the target scene, and no general pre-training is required. When the proxy geometry deviates much from the true shape, and/or the light transport effects are complex, more neural texture channels and a large neural rendering network are needed for full 360° free-viewpoint relighting, thereby imposing significant memory requirements that exceed the capabilities of current GPUs. We address the memory limitations with an effective view-partitioning strategy for the learned neural textures and neural rendering network, enabling the full capabilities of our relighting method on current generation GPUs. Our method is suited for capturing and relighting a scene with off-the-shelf handheld hardware (i.e., consumer cameras), thereby making full 360° free-viewpoint relighting practically accessible to a wider audience. We demonstrate the effectiveness of our method for scenes with a wide variety of material properties and global light transport effects. We show that high-quality results can be obtained for rough estimates of the geometry with a setup consisting of two handheld cameras. Both cameras capture the scene simultaneously while being lit by the flash light of only one of the cameras. Furthermore, we introduce a novel augmentation refinement strategy that exploits the linearity of light transport to generalize the relighting capabilities of the neural rendering network beyond the lighting conditions present during acquisition (i.e., flash lighting) to general environment lighting (Figure 1).

In summary our contributions are:

- a novel end-to-end system that enables full 360° free-viewpoint relighting from unstructured handheld captured photographs for a wide range of material properties and light transport effects;
- a deferred neural lighting renderer suitable for a wide range of lighting conditions;
- a novel handheld acquisition scheme that only requires two cameras; and
- an augmentation method for extending the relighting capabilities of our neural rendering network beyond the acquisition lighting.

2 RELATED WORK

We focus this overview of related work on methods that achieve one or both of our goals: multi-view rerendering and relighting.

2.1 Model-based Solutions

A classic tool for enabling multi-view rerendering is to explictly reconstruct the geometry of an object. Multi-view stereo [Seitz et al. 2006], visual hulls [Laurentini 2003], and photometric methods [Ackermann and Goesele 2015] are among the most versatile and popular methods. However, these methods only capture the shape of an object, and by themselves do not capture the view-dependent changes in appearance. Nevertheless, we will also leverage such methods, in particular multi-view stereo [Schönberger and Frahm 2016], to provide an estimate of the scene geometry. However, we do not expect perfect reconstruction accuracy, and only use the geometry estimate as a rough guide for rendering.

Appearance Modeling. Appearance modeling aims to capture the view and light dependent effects of surface reflectance (see [Weinmann and Klein 2015] for an overview). However, such methods require extensive probing of the appearance with various lighting conditions or require complex setups. While recent methods [Hui et al. 2017; Xu et al. 2016; Zhou et al. 2016a], including methods based on deep learning [Deschaintre et al. 2018, 2019; Gao et al. 2019; Kang et al. 2018; Li et al. 2017, 2018a; Ye et al. 2018], aim to reduce the number of required lighting conditions, these methods are limited to planar shapes or known accurate geometry.

Joint Modeling of Shape and Appearance. Jointly inferring shape and appearance is challenging due to the tight coupling of both

factors through the surface normals. Holroyd *et al.* [2010] capture both shape and appearance with a complex dual-arm gantry system. Xia *et al.* [2016] simplify acquisition by inferring shape and appearance jointly from a video of an object rotating under an unknown natural lighting environment. Similarly, Nam *et al.* [2018] reduce acquisition complexity by using photographs captured under active co-located point lighting. This reduction in acquisition complexity is traded off by a more complex, computationally expensive, and fragile, optimization process for estimating shape and appearance. Li *et al.* [2018b] leverage convolutional neural networks to robustly and efficiently infer shape and appearance from a single photograph; unseen surface points cannot be recovered, and due to biases in the shape combining multiple views is non-trivial [Vlasic et al. 2009]. Recently, Kang *et al.* [2019] learn optimal active lighting conditions, induced by a specially designed LED-cube, for jointly estimating shape and appearance. In concurrent work, Bi *et al.* [2020] estimate shape and reflectance from sparse multi-view images by jointly optimizing the latent space of the multi-view reflectance network to minimize the photometric error. However, geometry-based or BRDF-based solutions are inherently limited by the quality of the models. The quality of the reconstructions greatly depends on the completeness of the captured input, calibration accuracy, and accuracy of the model. Our method includes a neural rendering pass that aims to correct deficiencies in the shape and/or reflectance models.

2.2 Image-based Solutions

Image-based methods forego an explicit model of shape or appearance, and instead directly leverage the information embedded in images of the target.

Image-based Rendering. Light field [Levoy and Hanrahan 1996] and Lumigraph [Gortler et al. 1996] methods resample view rays from densely sampled views of the object. Leveraging prior knowledge of the shape improves view interpolation (e.g., via a global proxy geometry [Buehler et al. 2001; Chaurasia et al. 2013], or through view-dependent shape estimates [Hedman et al. 2018, 2016; Penner and Zhang 2017]). Surface light fields [Chen et al. 2018; Wood et al. 2000] aim to capture the changes in view-dependent appearance under high frequency lighting, such as point light sources, by storing a lumisphere for every point on the object's surface.

In the last few years, deep learning has been extensively explored as a means for improving novel-view-synthesis [Tewari et al. 2020], using various strategies ranging from flow-based warping methods [Jin et al. 2018; Liu et al. 2018; Park et al. 2017; Sun et al. 2018; Zhou et al. 2016b]), to view interpolation [Kalantari et al. 2016] and extrapolation [Srinivasan et al. 2019, 2017], to multi-plane images [Flynn et al. 2019; Mildenhall et al. 2019; Zhou et al. 2018], to tomographic volume representations [Lombardi et al. 2019], and to pure image-based disentangled learning [Ji et al. 2017; Olszewski et al. 2019; Yan et al. 2016; Yang et al. 2015]. Recently, Thies *et al.* [2019] demonstrated realistic view synthesis by jointly learning a deferred neural rendering network together with neural textures stored on an rough, inexact, proxy geometry. These neural textures encode the necessary features for the neural rendering network to correct inaccuracies in geometry and to correctly display view-dependent appearance. However, as with all of the above image-based rendering methods, the incident lighting is fixed at capture time. In contrast, our method allows for free-viewpoint rerendering of the scene under novel incident lighting by reshaping the neural textures not only based on a rough shape proxy, but also based on light-dependent radiance cues.

Image-based Relighting. In seminal work, Debevec *et al.* [2000] exploit linearity of light transport to reformulate scene relighting as a linear combination of photographs of the scene lit with different controlled lighting conditions. Subsequent work has focused on reducing storage requirements [Furukawa et al. 2002], accelerating relighting [Malzbender et al. 2001], reducing the number of required photographs [Peers et al. 2009], or acquisition under uncontrolled lighting either by limiting to specialized scenes (e.g., human subjects) [Guo et al. 2019; Li et al. 2013], landmarks [Haber et al. 2009], etc.), specialized lighting (e.g., outdoor natural lighting [Hauagge et al. 2014]), or simplified transport (e.g., lambertian reflectance [Imber et al. 2014]). Machine learning methods have been used to further reduce the number of required images for single view relighting [Meka et al. 2019; Ren et al. 2015; Sun et al. 2019; Xu et al. 2018] and very recently for multi-view relighting [Chen et al. 2020; Kanamori and Endo 2018; Meshry et al. 2019; Philip et al. 2019; Xu et al. 2019]. We will review this last category in more detail as it is closest related to our method.

Meka *et al.* [2019] learn to map two photographs of a human head under colored gradient illumination to a full 4D reflectance field. While at inference time only two gradient-lit photographs are needed, during training the full reflectance fields for 5 viewpoints of the subject are required. Kanamori *et al.* [2018] learn inverse rendering of ambient occlusion of full body photographs, assuming low frequency incident lighting and Lambertian surface reflectance. Both methods are specially geared towards human subjects and it is unclear how well these methods would extend to scenes with complex geometry (with cast shadows) and more general materials.

Mahmoud *et al.* [2019] demonstrate total scene rerendering of tourist landmarks from photo-collections using a proxy geometry in the form of a point cloud and formulate rendering as a multi-modal image-translation problem [Huang et al. 2018]. Consequently their rerendering method only offers an indirect control over incident lighting.

Philip *et al.* [2019] introduce a geometry-aware neural network that leverages geometry cues (e.g., normal maps and specular directions) and a rough geometric proxy to relight single-view inputs. Key to their method is a shadow refinement network (used on both source and target images). However, their method only models the incident lighting as a single directional light source (i.e., the sun) and an additional (ambient) cloudiness factor, making the method less suited for indoor scenes or general object relighting.

Xu *et al.* [2019] demonstrate high-quality multi-view relighting from a sparse set of wide-baseline photometric images under controlled lighting using 3D convolutions on a plane sweep volume aggregated with per-view per-depth attention maps. The relighting builds on Xu *et al.*'s earlier work on single-view relighting [Xu et al. 2018], and thus shares the same limitations. Importantly, it can only

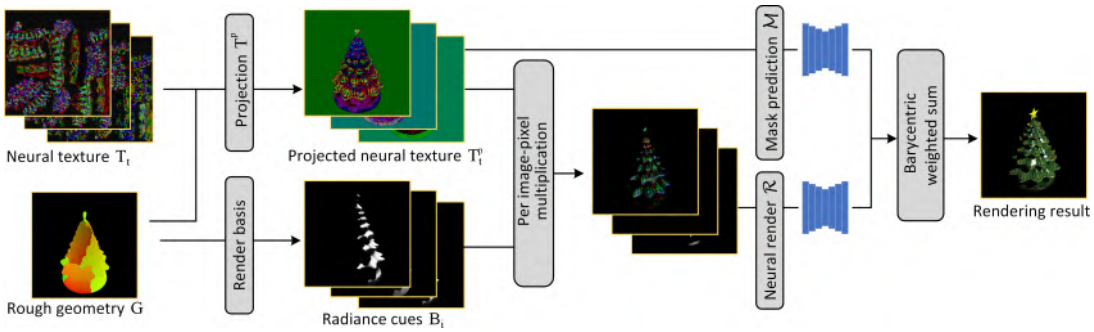

Fig. 2. Overview of Deferred Neural Lighting. First, an S-channel scene-dependent neural texture $\{\mathbf{T}_t\}_{t=1}^{S}$ is projected to a desired viewpoint $\mathbf{p}$ via a rough geometrical proxy $\mathbf{G}$ of the scene: $\mathbf{T}^p = \bar{\mathcal{P}}(\mathbf{T}, \mathbf{p}; \mathbf{G})$. Next, *radiance cues* $\{\mathbf{B}_i\}_{i=1}^{M}$ are synthesized by rendering M *scene-independent* basis materials $\{b_i\}_{i=1}^{M}$ under the target lighting $\mathbf{l}$ onto the rough geometry $\mathbf{G}$. Finally, the radiance cues and projected neural textures are combined (via a per-pixel multiplication) and passed to a scene-dependent neural rendering network $\mathcal{R}(\mathbf{T}_t^p \odot \mathbf{B}_i)$ that produces the final relit appearance of the scene. Additionally, to facilitate compositing the relit appearance, we also predict a binary mask from the projected neural textures.

relight from lighting directions from the frontal hemisphere, and thus is unable to relight a scene with full environmental lighting. Furthermore, Xu *et al.* rely on a dedicated acquisition setup, precluding in-situ capture. In contrast, we employ a more flexible handheld acquisition setup, and support the sphere of viewpoints and full environment lighting, at the cost of a denser viewpoint sampling.

In concurrent work and similar to us, Chen *et al.* [2020] are inspired by the deferred rendering pipeline of Thies *et al.* [2019] and encode a *"light transport function"* in the neural textures inferred from multi-view photographs of an object under unknown natural lighting. To regularize this is highly underconstrained problem, Chen *et al.* apply a number of heuristics and limit incident lighting to 10^{th} order spherical harmonic lighting. Consequently, their method cannot handle specular scenes and self-shadowing is typically omitted or baked in.

3 METHOD

3.1 Overview

Our algorithm takes as input a corresponding set $\{\mathbf{C}_k, \mathbf{p}_k, \mathbf{l}_k, \mathbf{M}_k\}_{k=1}^{N}$ of N photographs $\mathbf{C}_k$ of a scene with corresponding mask $\mathbf{M}_k$, intrinsic and extrinsic camera parameters $\mathbf{p}_k$, and incident lighting $\mathbf{l}_k$. We do not assume that the viewpoints or lighting conditions are distributed in a structured manner. Furthermore, we assume availability of a rough geometry $\mathbf{G}$, and a predefined set of M basis materials $\{b_i\}_{i=1}^{M}$.

As in Thies *et al.* [2019] we encode the view-dependent appearance of the scene by an S-channel learned neural texture $\{\mathbf{T}_t\}_{t=1}^{S}$ that lives in the UV texture space defined by the rough geometry $\mathbf{G}$. A neural texture acts similarly as a regular texture, but instead of storing appearance, normals, or displacements, a neural texel stores a learned S-length feature vector. These feature vectors will inform the (neural) renderer on how to compute the final pixel color. Given any view $\mathbf{p}$, we can compute the projection $\mathbf{T}^p = \bar{\mathcal{P}}(\mathbf{T}, \mathbf{p}; \mathbf{G})$ into the current viewpoint via the rough geometry. Unlike Thies *et al.* we do not directly pass the projected neural texture $\mathbf{T}^p$ to a neural rendering network $\mathcal{R}$, but instead embed lighting and material dependent information through a per-pixel multiplication of the projected neural texture channels $\mathbf{T}_t^p$ with radiance cues $\mathbf{B}_i$ (i.e., visualizations of the basis materials under the target lighting: $\mathbf{B}_i = \bar{\mathcal{R}}(b_i, \mathbf{p}_i, \mathbf{l}_i; \mathbf{G})$), which are then passed to the neural rendering network: $\mathcal{R}(\mathbf{T}_t^p \odot \mathbf{B}_i)$. Furthermore, we also predict a mask from the neural texture $\mathcal{M}(\mathbf{T}^p)$ which is post-multiplied with the output from the neural renderer. Figure 2 visually summarizes our algorithm.

As the neural texture informs the neural rendering network how to compute the output pixel values, and the neural renderer defines the exact meaning of the feature vectors, both the neural texture and the neural rendering network are trained in unison for each scene:

$$\mathbf{T}^*, \mathcal{R}^*, \mathcal{M}^* = \underset{\mathbf{T},\mathcal{R},\mathcal{M}}{\operatorname{argmax}} \sum_{i}^{N} \mathcal{L}(\mathbf{C}_i, \mathbf{p}_i, \mathbf{l}_i, \mathbf{M}_i | \mathbf{T}, \mathcal{R}, \mathcal{M}),$$

where $\mathcal{L}$ is a suitable loss function. We will describe each of the components that comprise our system in detail in the subsequent subsections.

3.2 Neural Textures

Neural textures for neural rendering were introduced by Thies *et al.* [2019], and we follow a similar end-to-end training and gradient update implementation. However, our work differs in three ways. First, in contrast to Thies *et al.* we deploy the learned neural textures differently, and hence they encode different types of appearance information. Second, unlike Thies *et al.*, we do not average over the different mipmap levels, but follow the standard usage of mipmaps. We use a 4-level mipmap hierarchy of neural textures. For each level, the mipmapping is computed as the average pooling from the previous level. Finally, we store more feature channels per neural texel (30), and hard assign the $i \times 6$ to $(i+1) \times 6$ feature channels to encode properties of the i-th basis material.

3.3 Deferred Neural Lighting

Deferred lighting typically computes a diffuse and specular light map. We generalize this to more "light maps" for deferred neural lighting. The core idea is, similar to how a rough proxy geometry

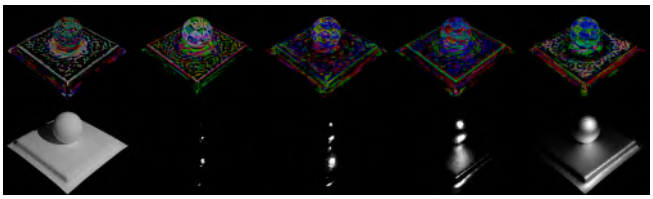

Fig. 3. Visualization of the projected neural textures (encoded in RGB images) and the corresponding radiance cue images for the synthetic *ball scene*.

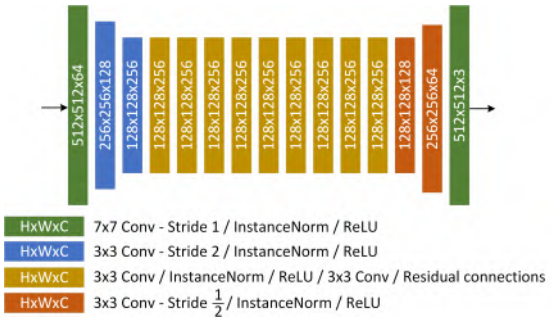

Fig. 4. Network structure of our neural rendering/masking network. A tanh activation is applied after the last layer, followed by exponentiation to undo the log encoding (Equation 1).

is leveraged in deferred neural rendering, to provide cues to the neural renderer on the impact of the incident lighting to the surface reflectance. We therefore include more than just two light maps, and leverage M "radiance cue" images, each rendered with a homogeneous basis BRDF. This is somewhat similar to the idea of representing surface reflectance with a set of basis materials albeit with a different goal [Ren et al. 2011]. Instead of characterizing the surface reflectance as a linear weighted sum of basis materials' reflectances, the neural renderer takes the radiance cues and combines them non-linearly, based on the neural texture information, into the final pixel values. This latter is also similar to Deep Shading [Nalbach et al. 2017], but instead of using a user-defined set of cues, we combine our cues with the learned neural textures.

Practically, we use $M = 5$ basis materials $\{b_i\}_{i=1}^{M}$, one characterized by a pure Lambertian BRDF, and the remaining 4 are modeled by the Cook-Torrance BRDF model [Cook and Torrance 1982] with roughness parameters $\{0.02, 0.05, 0.13, 0.34\}$ respectively. We use a GPU-based path tracer to synthesize the radiance cue images $\mathbf{B}_i = \bar{\mathcal{R}}(b_i, \mathbf{p}_i, \mathbf{l}_i; \mathbf{G})$, including indirect lighting, for each of the basis BRDFs b_i using the rough proxy geometry $\mathbf{G}$. Because the radiance cue images do not depend on learnable parameters, our GPU-based path tracer does not need to be differentiable. This greatly simplifies the learning process as well as the implementation. Note that the incident lighting $\mathbf{l}_i$ can be any type of incident lighting (e.g., directional light, environment lighting, etc.). Also note that the radiance cue images are 3-channel RGB images (e.g., to encode the effects of colored light sources).

Figure 3 shows the projected neural textures (encoded as RGB images), as well as corresponding radiance cue images for a synthetic specular *ball scene*.

3.4 Neural Rendering Network

The neural rendering network takes as input the per-pixel multiplied radiance cues and the projected neural textures: $\mathbf{T}_t^p \odot \mathbf{B}_i$. As noted before, we only multiply 6 neural texture channels with each radiance cue image. Since the radiance cue images are 3-channel RGB images, we interpret the neural texture channels as RGB too, thus yielding 2 RGB images (per radiance cue image) after multiplication (one for the first 3 neural texture channels, and a second for the last 3 neural texture channels).

Our neural rendering network follows the generator design of [Johnson et al. 2016; Zhu et al. 2017] with residual blocks [He et al. 2016]. We directly feed the multiplied radiance cue images with projected neural textures into the neural renderer. Our network structure is detailed in Figure 4. We found that a resblock based architecture more faithfully reproduces the appearance of the scene than the U-net based architecture used by Thies *et al.* [2019] (subsection 4.3).

To support a larger dynamic range, we further apply a log encoding after multiplying the neural textures and the radiance cues:

$$\frac{\log(x + \epsilon)}{\log(s)} + o, \tag{1}$$

where $\epsilon = \frac{1}{e}$ (with e being Euler's number), s and o are a normalization (exposure) scale and offset, respectively, that depend on the dynamic range of the input pixels x (with negative values clamped to zero) such that after log encoding the transformed values fall in the $[-1, +1]$ range. We also apply a log encoding to the training images; consequently computing the reconstruction loss operates in the log domain.

The neural renderer serves two goals simultaneous: it converts the multiplied radiance cues to pixel values, and it corrects errors in the rough geometry. When the rough geometry differs significantly from the actual shape or if the scene features complex light transport effects, then the neural renderer has to place more effort in correcting these errors, and thus a more capable network is needed, by for example increasing the number of neural texture channels. A practical issue with this approach is that the memory required for training exceeds the available GPU memory. We therefore take an effective alternative approach. Instead of training one neural renderer for a large number of neural texture channels, we partition the neural texture channels and train a dedicated neural renderer per partition. Since each partition is independent, we can train each neural texture partition plus corresponding neural renderer separately. We will show in subsection 4.3 that this partitioning does not adversely affect the accuracy of the neural renderer.

In practice, we use 13 separate neural networks, each with their own 30 neural textures and a neural rendering network. Each neural renderer is trained for a limited subset of view directions (but all lighting directions). To determine the subset of view directions, we uniformly distribute 13 vertices on the sphere of view directions (approximately 60° apart), and triangulate these vertices. Each vertex corresponds to a neural rendering network, and it is trained on all view directions that fall within the adjacent triangles. Thus, for each

view direction through a triangle, we have three predicted renders (from the neural networks associated with the triangle's vertices) that we blend based on barycentric weighting to the final relit result.

3.5 Neural Mask

To aid in compositing the relit object on a new background, we also estimate a binary mask given the viewpoint. We use a neural network to estimate the mask directly from the projected neural textures. The neural mask network follows the same structure as the neural rendering network with only half the number of intermediate channels, outputting only one channel mask via a sigmoid activation instead of a tanh activation after the last layer.

3.6 Training & Data Capture

We train our network end-to-end on a set of unstructured photographs from the same scene using an ℓ_1 loss between the log prediction and the corresponding (log) photograph, and a cross-entropy loss for the mask; both losses are weighted equally. We implemented our method in TensorFlow [Abadi et al. 2015] and train our network with the Adam optimizer [Kingma and Ba 2015], with a 0.0002 learning rate, using a batch-size of 1, and set $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We train each partition's network on an NVidia P100 GPU for 20 hours. An additional advantage of partitioning the neural texture with associated neural renderers, is that we can trivially parallelize training over multiple GPUs.

We will demonstrate our deferred neural lighting method with a handheld capture setup using mobile phone cameras as well as DSLR cameras. Our default setup utilizes two cameras, in video mode, moved independently around the scene. One of the cameras also has its (co-located) flash light turned on; we assume no other major sources of incident lighting are present. We aim to get a good coverage of possible viewpoints and cover a wide variety of lighting directions (flash-on camera) for each nearby view (flash-off camera). While multi-view stereo can be used to estimate the camera locations of each camera, we place a checkerboard in the scene to aid estimating the camera parameters. Since the light source (i.e., flash light) is co-located with one of the cameras, we can use the corresponding extrinsic camera parameters as the location of the light and model it as a point light. This avoids the geometric and radiometric calibration between the two cameras. We employ a gamma 2.2 correction to transform the recorded pixel values to (approximately) radiometrically linear measurements. Although it is possible to use the captured frames from both cameras, in practice, we found using the captured frames from the flash-off camera are sufficient to train our neural relighting system. We refer to the supplemental video for a (sped-up) capture sequence.

We employ COLMAP [Schönberger and Frahm 2016] for reconstructing the rough geometry from photographs of the object under fixed natural lighting (i.e., without using the cameras' flash lights). While geometry reconstruction is possible from the co-located images, the non-stationary specular highlight affects the quality significantly. Figure 5 shows the estimated rough geometries for each of the scenes used in this paper. We perform minimal manual cleaning to the recovered COLMAP geometry; we remove features outside the region of interest by either selecting the largest continuous shape

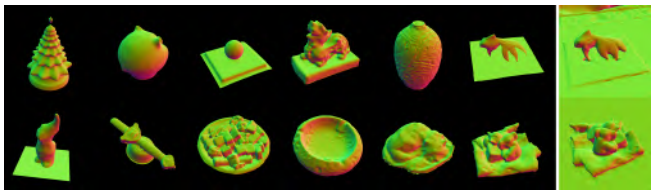

Fig. 5. Visualizations of the rough proxy geometries, color-coded by the surface normal directions, used for the different real and synthetic scenes. The right most column shows two examples of the raw COLMAP geometry to illustrate the degree of manual intervention.

or by providing a bounding box (Figure 5, right column bottom example). In the case when COLMAP partially fails, we manually fix the affected region; this only occurred for the Sphere and Fish example (Figure 5, right column top example). We employ *closed form matting* [Levin et al. 2008] to compute the masks based on trimaps computed by shrinking/growing the silhouettes of the projected rough geometry.

We captured video sequences of 9 scenes to demonstrate the effectiveness of our method, as well as simulated captures of 3 synthetic scenes for validation (following the real capture process as close as possible using exactly the same calibration and reconstruction process). The captured scenes are: a *gnome* on a glossy surface (with glossy interreflections), a *bronze vase* (with rough specular reflections), a *candy bowl* (with intricate shadowing effects not captured by the rough geometry), an *empty bowl* (with large scale occlusions), a *decorative sword* (with fine texture detail), an *ornamental fish* statue (with fine texture details), a *Pixiu statuette* (with translucency), a *cat* (with fur), and a *cluttered scene* with a variety of objects (displaying complex shadowing, anisotropic reflections, and interreflections). The *ornamental fish*, *Pixiu statuette*, the *cat*, and the *cluttered scene* are captured with a DSLR camera pair, while the other scenes are captured with a regular mobile phone setup. The synthetic scenes exhibit a wide variety in geometric details (e.g., the back side of the *pig head* scene), challenging material properties (e.g., a textured specular *ball scene* including a rough specular ground plane), complex shadowing (e.g., the *Christmas tree*), texture detail (e.g., the *ball scene* and the *Christmas tree*), and strong interreflections (e.g., the reflections in the *ball scene*).

For the synthetic scenes we sample 10,000 views, each with different light source positions. The number of captured video frames used for the real examples shown in this paper varies per scene between 6,000 and 20,000 views (see Table 1, first column, for a summary of the number of captured views). While the number of captured views might appear high, especially compared to prior SVBRDF modeling methods (e.g., [Nam et al. 2018]), we remark that our method is an *image-based* neural relighting method which typically requires more captured photographs but, compared to SVBRDF methods, can handle more intricate scenes with a richer variety in materials and more complex light transport. To put this in perspective, lets assume we use classic image-based rendering for view interpolation from relit views generated by a fixed-view relighting method. In that case, we would only have 100 lighting directions for 100 views, which would be challenging to view-interpolate or

relight with many image-based methods. Furthermore, we extract frames from a video sequence, which would ideally take less than 6 minutes on a 30fps video camera. Practically, capturing times are longer (15 ∼ 30 minutes) because the flash light on the camera is relatively dim, and thus to avoid motion blur, the cameras need to move slowly (and we subsample the frame rate to get a rich variation of views). However, brighter flash lighting and faster frame rates can potentially reduce acquisition time significantly.

Figures 6 and 7 compare results for each of the scenes visualized from a novel viewpoint and lit from a novel lighting direction with a reference photograph not used for training. Overall, the relit results are visually a good match to the reference photographs. Please refer to the supplemental material for visualizations with different view and lighting combinations.

3.7 Lighting Augmentation

Our deferred neural lighting method is agnostic to the type of incident lighting used for rendering the radiance cues. Hence, it naturally supports different types of lighting ranging from local to distant lighting, and from directional lighting to environment lighting. However, attempting to relight the scene with environment lighting does not necessarily yield a plausible relighting because the neural rendering network is trained exclusively on white point lighting. Consequently, the rendering network has no concept of area or colored light sources as shown in Figure 8.

To address the lighting generality issue, we exploit linearity of light transport and perform an additional augmentation training refinement pass. The key idea is to refine the network with novel training images of the scene lit by environment lighting generated through classic image-based relighting [Debevec et al. 2000] using basis images under directional lighting predicted by the neural network itself. However, naively applying this augmentation step is impractical as this would require generating for many viewpoints (e.g., 1,500 views) a large number of images (e.g., for an environment map encoded as a $32\times32\times6$ cube map, this yields 6,144 basis images per viewpoint). We therefore employ importance sampling on the environment lighting and approximate the relighting using only 100 light samples for each of the 5 light probes and for each of the 1,500 selected views (yielding a total of 7,500 augmented training samples). We train the network with an even mix of augmented training images and the original captured images. We found that 5 training light probes, each randomly selected for each training viewpoint of the scene from a set of 90 light probes, is sufficient to generalize to other natural light probes not part of the training set. Figure 9 shows relighting results under environment lighting (not part of the 90 training light probes) and the corresponding reference ground truth results for the synthetic scenes. Figure 1 includes augmented relit results for selected captured scenes. For scenes with strong specular reflections, using 100 light samples is not always sufficient for the neural rendering network to learn how to handle low frequency ambient lighting. In such cases (e.g., the specular *Ball scene*), we use an exhaustive relighting with all light directions. Data preparation for lighting augmentation is costly and takes about 5 hours per partition, and 1.5 hours for relighting, and an additional 20 hours of training per partition using the previously trained network as a starting point.

Table 1. Quantitative evaluations of relighting results for each synthetic and captured scene. The respective absolute errors (AE) and perceptually-based LPIPS errors are computed over 1,000 view/lighting combinations for the captured scenes, and 1,384 view/lighting combinations for the synthetic scenes.

	Tot.	Mean		Maximum	
	#Input	AE	LPIPS	AE	LPIPS
Pig head	10,000	0.0030	0.061	0.0055	0.160
Sphere (Specular)	10,000	0.0007	0.003	0.0012	0.013
Sphere (Diffuse)	10,000	0.0006	0.017	0.0016	0.039
Sphere (Mixed)	10,000	0.0014	0.035	0.0059	0.072
Christmas tree	10,000	0.0017	0.043	0.0040	0.099
Candy bowl	16,729	0.0089	0.051	0.0160	0.084
Bronze vase	17,024	0.0017	0.037	0.0059	0.092
Gnome	14,132	0.0034	0.032	0.0110	0.056
Empty bowl	19,682	0.0034	0.046	0.0094	0.066
Decorative sword	13,537	0.0024	0.015	0.0052	0.029
Ornamental fish	13,032	0.0039	0.121	0.0170	0.180
Cat	6,389	0.0019	0.018	0.0037	0.029
Pixiu statuette	13,928	0.0036	0.066	0.0061	0.130
Cluttered scene	16,720	0.0040	0.066	0.0092	0.089

4 DISCUSSIONS

4.1 Validation

Figures 6 and 7 demonstrate that our method is able to achieve visually plausible results. To further validate our results, we quantify the maximum and mean Absolute Error (AE) and the perceptually based LPIPS [Zhang et al. 2018] error between 1,000 reference and predicted images with random view and lighting direction (not part of the training dataset) for both the real and synthetic scenes. The results are summarized in Table 1, showing that our method achieves high accuracy. The higher AE errors on the *Candy bowl* are due to artifacts in the mobile phone captured frames (i.e., motion blur, defocus blur, and sensor noise). The higher LPIPS error for the *Ornamental fish* scene are due to the glint-like texture on the ground plane which is challenging to sample and reconstruct exactly.

4.2 Comparison To Prior Work

A direct comparison to prior work is difficult as there currently does not exist a method that can relight under exactly the same conditions (i.e., unstructured photographs, complex material properties and light paths, and full 360° relighting). We therefore make a best effort comparison with methods that solve similar problems.

The method of Nam *et al.* [2018] reconstructs shape and spatially varying BRDFs from handheld captured backscatter observations. However, their method ignores the impact of interreflections, which can adversely affect the results for scenes with strong interreflections, as shown in Figure 10. The SVBRDF estimation (starting from the same rough geometry as our method) and rendering (with direct lighting only) were kindly provided by Nam *et al.*

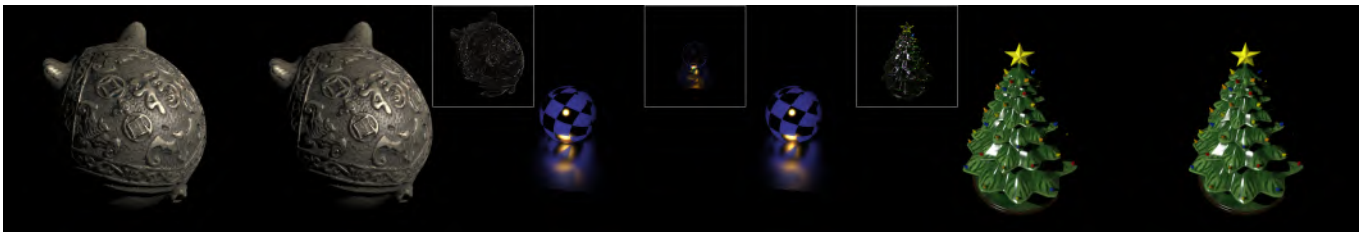

Fig. 6. Qualitative comparison between synthetic scenes relit (right) for a novel viewpoint and lit from a novel lighting direction (not part of the training data) and a rendered reference photograph (left). Difference images (×5) are shown in the insets.

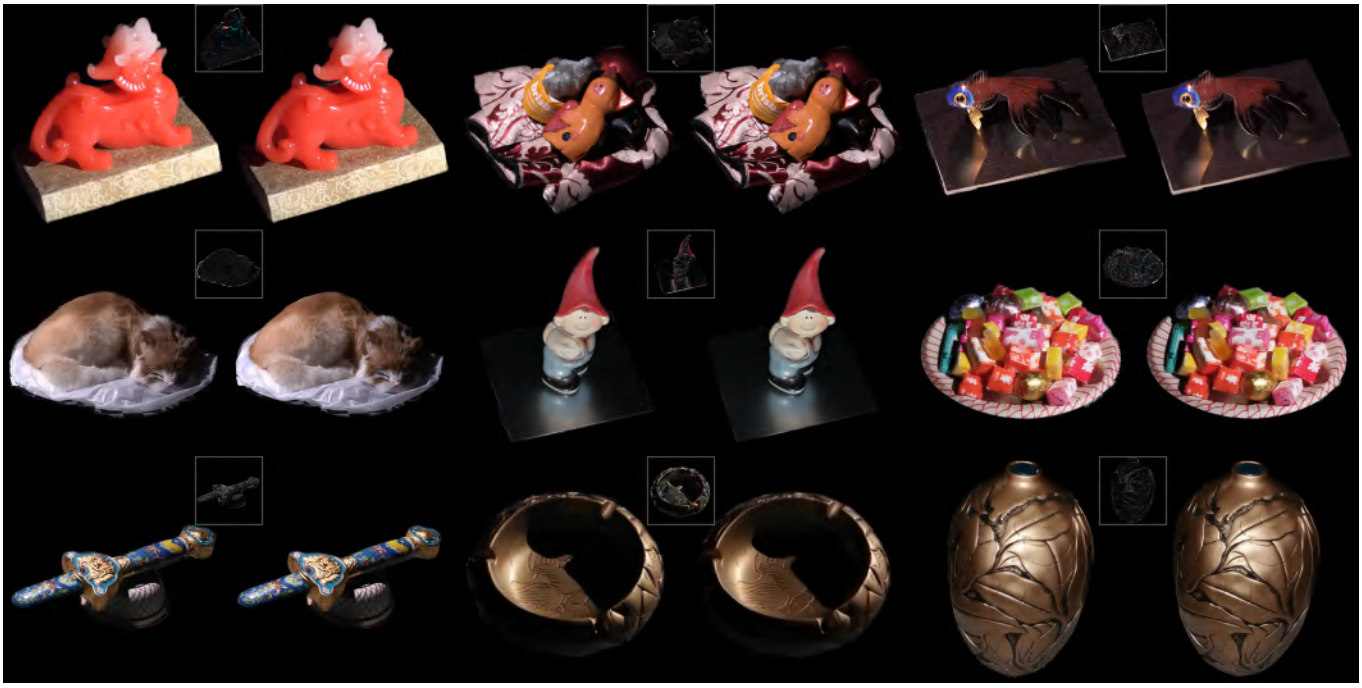

Fig. 7. Qualitative comparison between captured scenes relit (right) for a novel viewpoint and lit from a novel lighting direction (not part of the training data) and a captured reference photograph (left). Difference images (×5) are shown in the insets.

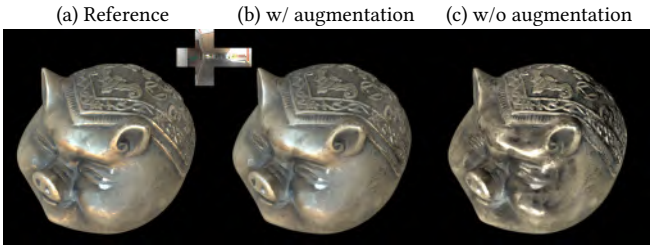

Fig. 8. Without lighting augmentation, the neural network trained with only point light source lighting has no concept of area or colored light sources, yielding artifacts when attempting to relight with environment lighting (c) compared to the reference (a). Our lighting augmentation generalizes the neural networks to enable relighting with environment lighting (b).

High quality *single-view* relighting was demonstrated by Xu *et al.* [2018], and the same method served as basis for their multi-view follow-up work to enable relighting [Xu et al. 2019]. In contrast to our method, their relighting network generalizes to other scenes. However, this comes at the cost of only being able to relight from the frontal hemisphere and it has difficulty handling complex long-range lighting effects, such as the interreflections between the ball and the ground plane as shown in Figure 10.

Deferred Neural rendering *et al.* [Thies et al. 2019] also relies on neural textures. However, a key difference between our method and the deferred neural rendering method of Thies *et al.* is that they provide viewpoint information to the neural rendering network by multiplying the lowest 3 spherical harmonics bands with 9 of the projected neural textures. Given the prominence of spherical harmonics in precomputed radiance transfer [Sloan et al. 2002] and inverse rendering [Ramamoorthi and Hanrahan 2001], this raises the question whether a direct augmentation of deferred neural rendering in which the lighting is embedded via spherical harmonics in a

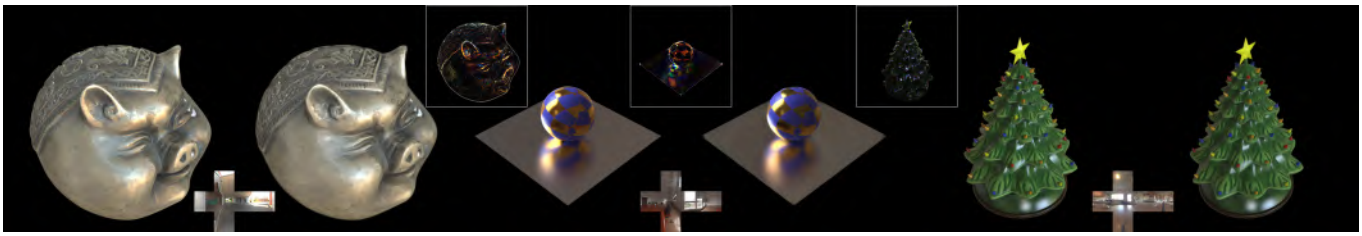

Fig. 9. Qualitative comparison between synthetic scenes relit for a novel viewpoint by the light probes shown as insets at the bottom. Difference images (×5) are shown in the insets at the top.

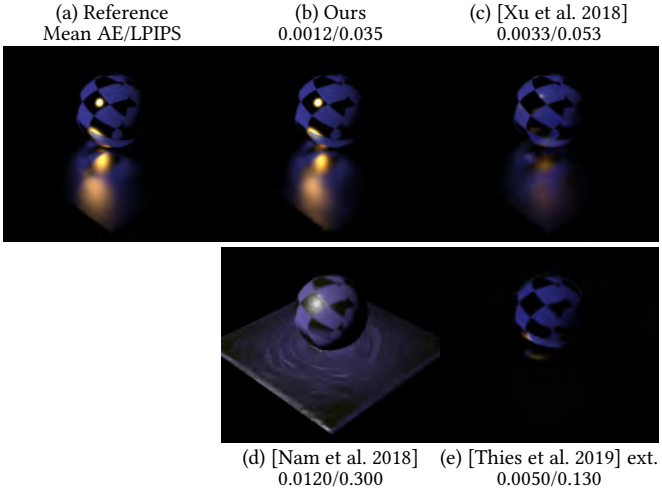

Fig. 10. Comparison to selected prior work. Our method is able to plausibly reproduce the interreflection effects for this challenging scene, while the learning-based relighting method of Xu *et al.* [2018] fails to fully capture the non-local light transport effects. Similarly, the quality of the results of the SVBRDF and geometry reconstruction method of Nam *et al.* [2018] is adversely affected by the strong interreflections. Naively extending deferred neural rendering [Thies et al. 2019] by encoding the light source direction with Spherical Harmonics coefficients fails to reproduce the global illumination effects.

similar fashion (using 9 additional neural texture channels) would yield a relightable deferred neural rendering solution. Figure 10 (e) show that such a naive extension trained on the same images as previous results fails to generalize to unseen lighting directions. It is important to note that the neural renderer attempts to learn the relation between lighting direction (expressed in 9 spherical coefficients) and the observed reflectance in the training images (under a single light source). Consequently, the use of just 9 spherical harmonics coefficients does not imply that it is naturally limited to diffuse surface reflectance only.

4.3 Ablation Study

Impact of Neural Rendering Architecture. The combination of neural textures and neural rendering was introduced by Thies *et al.* [2019] for, among others, view-interpolation. However, we found that the neural renderer architecture of Thies *et al.* did not produce as accurate results when used for deferred lighting as demonstrated in Figure 11 (c) and which furthermore resulted in temporal artifacts (e.g., screen door effects, shimmering, etc.) when changing viewpoint. Instead we use a more powerful generator design with residual blocks, which more faithfully reproduces the appearance (Figure 11(b)).

Impact of Number of Basis Materials/Radiance Cues. Classic deferred lighting only computes diffuse and specular reflectance. However, unlike our method, it has exact knowledge of the material properties. Similar as how deferred neural rendering is robust to inaccuracies in the geometry, our deferred neural lighting method only requires rough estimates of the reflectance (i.e., "cues"). In Figure 11 (d-f) we explore the impact of altering the number of basis materials/radiance cues on the synthetic scenes, while keeping the number of neural texture channels per basis material constant. As can be seen, increasing the number of basis materials (f) provides marginal benefit, whereas decreasing the number of materials (d-e) results in a significant reduction in quality. Therefore, we opt for using 5 basis materials as this strikes a balance between relighting quality and cost of evaluation (i.e., each additional radiance cue imposes a rendering cost). The quantitative mean AE and LPIPS errors in Table 2 further confirm our observations.

Impact of Number of Training Photographs. Figure 11 (j-l) demonstrates the impact of the number of training photographs on visualizations for the synthetic *Christmas tree* scene trained with 500, 1,000, and 2,500 photographs sampled from the same sequence of 10,000 input photographs (b). We can see that starting from 2,500 input photographs, a plausible relighting can be obtained for a fixed viewpoint. Increasing the number of input photographs improves relighting accuracy and exhibits more details. However, when changing viewpoint, we observe temporal instabilities that decrease with increasing number of input photographs. Crucially, we do not observe such temporal instabilities when changing the lighting. This suggests that the majority of the input photographs are needed to correct inaccuracies in the proxy geometry and to support free-viewpoint rendering. The mean AE and LPIPS errors (Table 2) further show that with increasing number of input photographs, a lower error is obtained.

Impact of Number of Neural Texture Channels. A second important parameter in our deferred neural lighting method is the number of neural texture channels assigned to each radiance cue. As shown

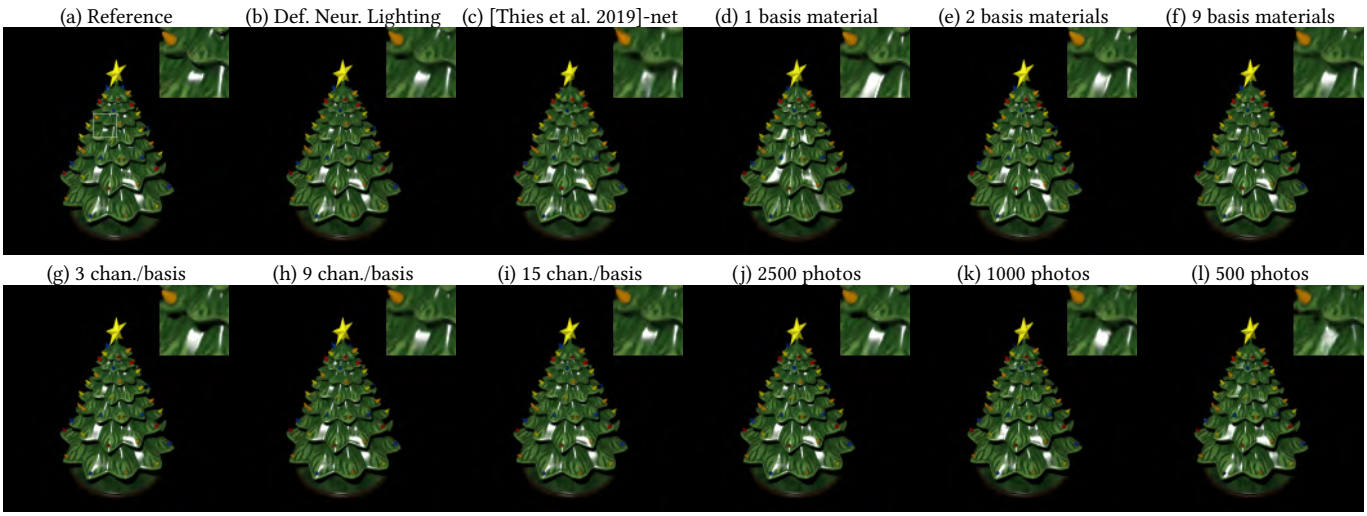

Fig. 11. Ablation study of deferred neural lighting. Compared to a reference visualization of the synthetic *Christmas tree* scene (a), our method produces plausible relit results (b). Using the neural rendering network architecture of Thies *et al.* [2019] fails to reproduce all specular highlights faithfully (c) and introduces temporal artifacts when changing the view. Key to our method's ability to relight scenes are *radiance cues*. A single diffuse basis material (d) (and thus a radiance cue that encodes incident irradiance) does not yield an accurate relit result (although it is able to reproduce some specular highlights). While better, adding a specular radiance cue (e) (similar to classic deferred lighting) does not faithfully reproduce the appearance. We found that 5 basis materials strike a good balance between quality and evaluation cost; increasing the number of basis materials further (f) only provides marginal improvements. The number of neural texture channels per radiance cue also impacts visual quality; decreasing the number of channels yields some loss of quality (g), whereas an increase provides modest improvement (h,i). We opt for using 6 neural texture channels per radiance cue, striking a balance between training time and accuracy. Finally, the number of training photographs (each from a novel view and under a novel lighting direction) has significant impact on the relighting quality (j-l). We found that, for typical proxy geometry quality, plausible results can be obtained for fixed viewpoint relighting starting from 2,500 photographs, and that 10,000 photographs further yields visually more stable results when changing the view.

Table 2. Quantitative evaluation for the ablation experiments averaged over all three synthetic scenes. The respective errors are computed over 1,384 view/lighting combinations not part of the training set. Our default settings are highlighted in bold.

Ablation Variant	MAE	LPIPS
Using [Thies et al. 2019] network structure	0.0042	0.073
Small partition range (30°)	0.0039	0.045
Medium partition range (60°)	**0.0029**	**0.038**
Large partition range (90°)	0.0042	0.049
1 basis material	0.0057	0.049
2 basis materials	0.0038	0.044
5 basis materials	**0.0029**	**0.038**
9 basis materials	0.0029	0.038
3 channels per basis material	0.0031	0.038
6 channels per basis material	**0.0029**	**0.038**
9 channels per basis material	0.0030	0.038
15 channels per basis material	0.0028	0.037
500 training photographs	0.0049	0.053
1,000 training photographs	0.0042	0.045
2,500 training photographs	0.0032	0.040
10,000 training photographs	**0.0029**	**0.038**

in Figure 11 (g-h), the number of texture channels impacts the visual accuracy. However, in terms of quantitative errors, we only see a modest improvement with increasing number of neural texture channels per radiance cue image (Table 2). Interestingly, note that using only 3 channels per basis material uses approximately the same number of neural texture channels as deferred neural rendering [Thies et al. 2019], while providing additional relighting functionality. In our implementation, we opt for 6 (i.e., 2× RGB) neural texture channels per basis material, striking a balance between accuracy, training time, and inference efficiency.

Impact of Neural Texture Partitioning. In subsection 3.4 we introduced our neural rendering architecture and indicated that, due to memory constraints, we partition the neural texture channels and independently train a dedicated neural rendering network for each partition. To better understand the impact of this choice, we compare a number of alternative rendering schemes:

(1) We refer to the *Monolithic Architecture* as the solution without any partitioning and using the same number of neural texture channels as the total of all neural texture channels over all partitions.
(2) *Joint Training* is similar to our partitioned solution, except that we train all of the neural textures and neural renders jointly and the loss function is evaluated on the interpolated results. This contrasts to our solution where we train each partition and neural renderer separately.
(3) For the *Shared Neural Renderer* scheme we share the same neural renderer between all partitions.

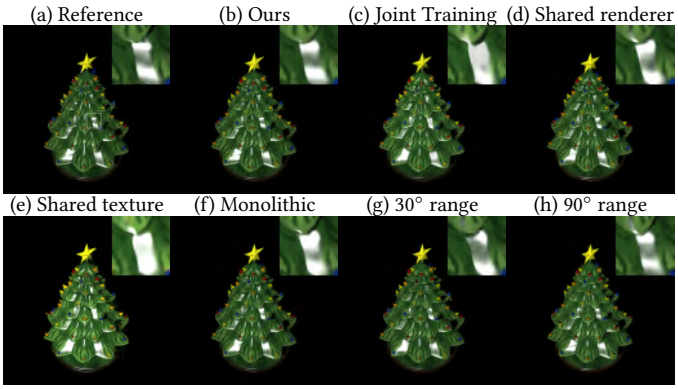

Fig. 12. Comparison of the relighting quality on the synthetic *Christmas tree* scene with alternative rendering schemes. We found that our partition solution (b) trained independently faithfully reproduces specular highlights. (c) Joint training yields degraded results, suggesting that each individual partition should be optimized separately. The results from sharing either the neural renderer (d) or the neural texture (e), indicates that the partitioning of the neural texture plays a more important role in render quality. Training a monolithic network (f) produces results of similar quality as our partitioned solution, but at the cost of much larger memory requirements and computational cost. For a fixed total number of neural texture channels, increasing the number of partitions (g) produces less accurate results due to a reduction in total number of training exemplars per partition. Decreasing the number of partitions (h) produces a less accurate results, and exhibits artifacts when changing viewpoints.

(4) For the *Shared Neural Textures* scheme we share the neural textures between the partitions.

All these alternatives exceed the memory constraints of current GPUs. To maintain a comparable light transport complexity while reducing memory consumption, we only train these alternative schemes for a view range equivalent to 3 partitions (i.e., a single triangle) instead of a full hemisphere of partitions as in our implementation.

Figure 12 provides a qualitative comparison on the synthetic *Christmas tree* scene between the different neural rendering schemes. Our independently trained solution (b) is visually close to the reference (a). The *Joint Training* (c) solution shows significant differences in the highlights. Furthermore, we observe that the *Shared Neural Renderer* produces more accurate renditions than the *Shared Neural Textures*, indicating that partitioning the neural texture is essential. Finally, the *Monolithic Architecture* performs similarly to our independently trained solution, indicating that our solution is a viable alternative. The quantitative errors listed in Table 3 agree with the qualitative conclusions.

Impact of Number of Partitions. In our implementation we used 13 partitions which roughly corresponds to a 60° separation between vertices. The number of partitions depends on two interdependent factors: first, how efficiently can the appearance be modeled by the selected number of neural texture channels, and second how many training samples (i.e., captured frames) can be used for training. The number of texture channels essentially determines the upperbound on the separation angle. For example, when going to 90° separation between vertices, and keeping the total number of neural texture channels and training samples fixed, reduces the rendering quality (Figure 12(h)) and introduces visual artifacts when changing viewpoints. Reducing the separation angle and thus increasing the number of partitions, reduces the number of training images per partition (when keeping the total fixed), and thus also the lighting variations seen during training yielding a less accurate neural renderer (Figure 12(g)).

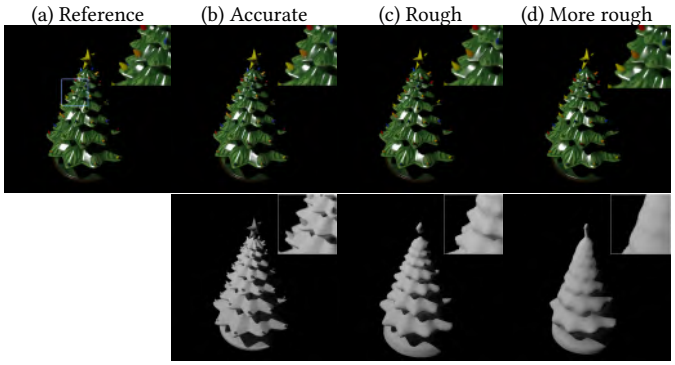

Fig. 13. Comparison of the relighting quality on the synthetic *Christmas tree* scene for different levels of accuracy for the proxy geometry. With accurate geometry, our method can recover detailed appearance effects from just 1,000 photographs, providing a close match to the ground truth (a). Using a proxy geometry quality similar to that obtained from real-world acquisition, yields good reconstruction of all the appearance details and shadows at 2,500 photographs. For an even more rough geometry proxy, our method can still produce plausible relit results, but at the cost of more training photographs (10,000). Note that the geometry of "star" on top of the tree is severely degraded in (c) and (d). Yet, our method is able to correct these missing features and faithfully synthesize its (relightable) appearance.

Table 3. Quantitative evaluation for the neural texture partitioning experiments on the Christmas Tree scene. The respective errors are computed over 180 view/lighting combinations not part of the training set. The lowest error is marked in bold.

Ablation Variant	MAE	LPIPS
Monolithic Architecture	**0.0026**	0.059
Joint Training	0.0036	0.067
Shared Neural Renderer	0.0029	0.063
Shared Neural Textures	0.0040	0.071
Independent partition	0.0027	**0.056**

Impact of Geometry Accuracy. Similar to deferred neural rendering [Thies et al. 2019], our deferred neural lighting is robust to geometric errors. All synthetic examples shown in this paper are created with rough geometry of similar quality as expected from real-world acquisition (Figure 13 (c), trained on 2,500 photographs). However, our method can still produce good results for even more rough geometries (Figure 13 (d), trained on 10,000 photographs). Furthermore, higher geometry accuracy also helps in recovering small specular highlights (Figure 13 (b), trained on 1,000 photographs). As can be seen, the quality of the geometry is closely tied to the number

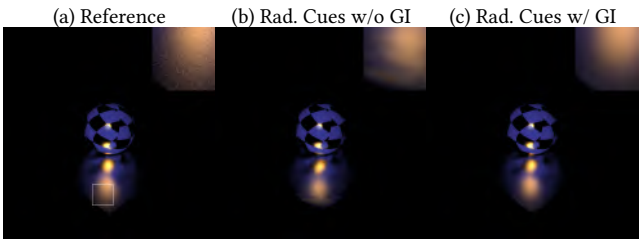

Fig. 14. Comparison between relighting results from radiance cues rendered with and without indirect lighting (and using an appropriately trained corresponding neural rendering network). Our method is able to recover a significant portion of the indirect lighting even if the radiance cues are not rendered with global illumination effects. However, including indirect lighting in the radiance cues produces more accurate relit results.

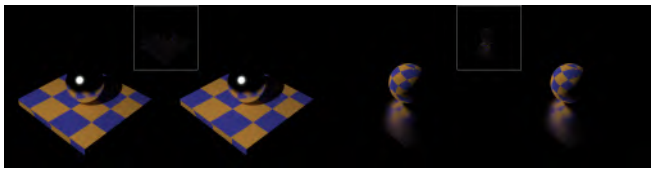

Fig. 15. Qualitative validation of the neural renderer's ability to correctly relight interreflections between different materials. 1st example: a glossy sphere on a diffuse spatially varying ground plane. 2nd example: a diffuse sphere on a glossy ground plane. Each example is relit from a novel viewpoint and lit from a lighting direction not part of the training (shown on the right) and compared to a reference visualization (shown on the left). The differences images (×5) show that the indirect lighting is faithfully reproduced.

of required viewpoints; a better geometry allows for longer-range interpolation of view information. However, the exact relation between geometrical accuracy and number of views highly depends on the complexity of the shape and material properties.

Global Illumination vs. Local Shading in Radiance Cues. In all our results we employ path tracing to produce the radiance cues including global illumination effects. The key idea is that the neural rendering network can take these indirect cues and transform them into correct indirect lighting in the relit images. Since the neural renderer already has to correct these cues, a natural question arises on whether the indirect lighting in the radiance cues is necessary. Figure 14 shows a challenging synthetic scene with strong indirect lighting effects relit using radiance cues with and without indirect lighting. We observe that our neural representation is able to plausibly predict the majority of the indirect lighting even if no indirect lighting was present in the radiance cues. However, we observe that the quality is lower than those produced with radiance cues with indirect lighting.

The previous experiment shows that including indirect lighting in the radiance cues helps in reconstructing plausible interreflections even for spatially varying materials, including translucency, despite the fact that the radiance cues are computed from homogeneous opaque materials. Figure 15 further illustrates the capabilities of our neural deferred lighting network to reproduce complex interreflections between different materials despite the fact that our radiance cues are generated with a single material per cue. The ability to reconstruct complex interreflection between materials that differ significantly from radiance cue materials is made possible by the same capability that: allows our deferred neural lighting method to introduce indirect lighting effects that were not present in the radiance cues due to missing geometrical features in the rough geometry, correct the reflectance due to incorrect normals, and add missing geometrical details. All these operations *non-linearly transform* the possibly incorrect direct lighting, incorrect indirect lighting, and shape encoded in radiance cues (times the neural textures) to plausible renderings. As shown in our ablation study, the more correct the shape or the radiance cues, the easier it is for the neural renderer to produce accurate results. An interesting avenue for further investigation would be to explore different encodings of the radiance cues, e.g., by encoding direct and indirect lighting separately.

Radiance Cues. We have opted to combine the radiance cues with the neural texture by multiplication. As noted, this process was inspired by how surface reflectance of spatially varying materials is often modeled as a linear weighted sum of basis materials. We also experimented with other strategies, such as concatenating the radiance cues to the feature vectors before passing them into the neural renderer. However, we found that this resulted in less stable training and lower quality results. We suspect that by multiplying, the radiance cues are explicitely coupled to a fixed subset of the neural textures, thereby ensuring an even distribution. Concatenating, on the other hand, does not enforce this, and might result in a suboptimal distribution. An interesting avenue for future work would be to investigate different strategies for combining the cues and the neural textures.

4.4 Limitations

Our method is not without limitations. We observe that small and narrow highlights are not always reproduced. This is likely due to two underlying reasons. First, small highlights induce a small localized error, and hence are more difficult for the network to learn. Second, we observe that the quality of the proxy geometry plays a significant role; a more accurate geometry yields a more accurate reproduction of such small and narrow specular highlights as demonstrated in Figure 13. This argument is further strengthened by the observation that missing highlights in the relit images typically also do not show up in the radiance cue images.

While the neural rendering network is able to correct inconsistencies in shape between the radiance cues and the target images, the effect of camera calibration errors impacts both components, and hence cannot be corrected (i.e., an incorrect viewpoint estimate will produce a radiance cue seen from an incorrect viewpoint. However, the corresponding target view will also be seen from the incorrect view). Consequently, our method is sensitive to camera calibration errors. Visually, this translates into *"wobbling"* when the viewpoint is changed as the network attempts to reproduce the *incorrect* camera calibration.

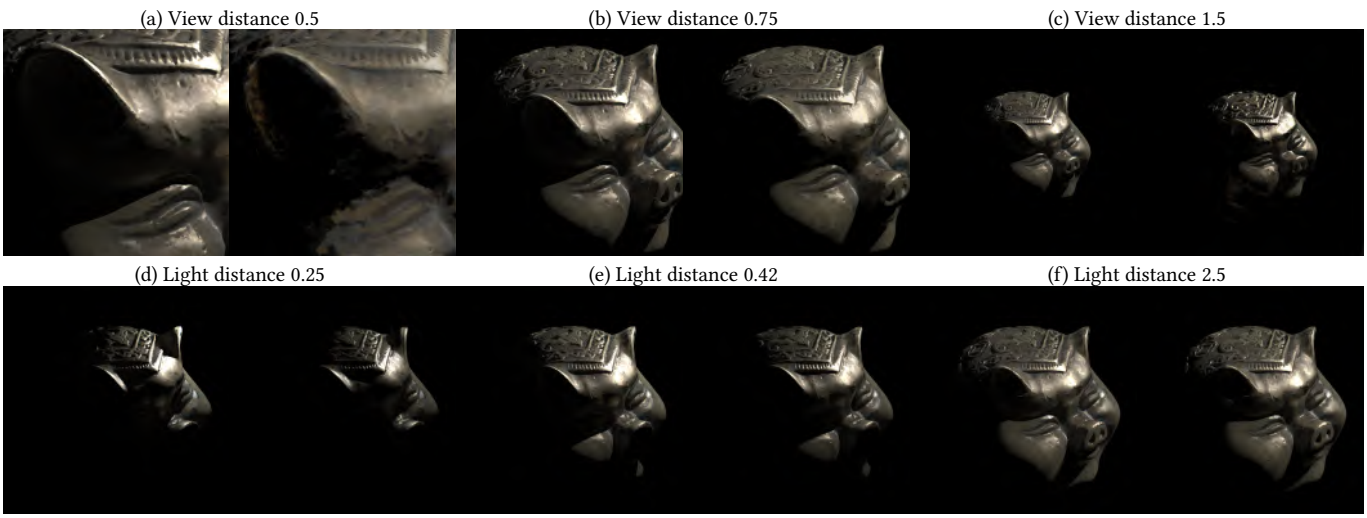

Fig. 16. Examples of the impact of view and lighting distance demonstrated on the synthetic *pig head* scene (left: reference; right: relit result). Our deferred neural lighting method is robust to moving the light and view farther away from the object (c,f) than the original capture distance (i.e., view and light distance 1.0). While our method can handle some degree of inward motion of view and lighting (b,e), it fails to produce plausible results for extreme near-field view and light positions (a,d).

Furthermore, while our method is fairly robust to inaccuracies in the proxy geometry, it is also not without limitations. In particular we observe that large missing parts in the geometry pose a challenge to our method.

The accuracy of our lighting augmentation strategy is limited by the accuracy of the relit basis images obtained with deferred neural relighting trained on unstructured photographs with flash lighting. One of the most significant sources of error is when the training photographs do not cover the full dynamic range of the scene; this is common for scenes with strong specular highlights. In such a case, our method cannot reproduce the full dynamic range either, and thus the synthesized image-based relit training images will be incorrect. Furthermore, we currently employ importance sampling with 100 samples to reduce the cost of synthesizing the augmented training samples. However, this assumes that the 100 light samples can accurately approximate the relit appearance of the scene.

As noted in subsection 3.3, the radiance cues can be generated from any view and for any lighting condition. However, the neural rendering network can only render images that are in the space covered by the training photographs. Experimentally, we found that our method behaves well for viewpoint and light positions farther away than those used in acquisition (Figure 16 (c,f)). We also found that we are able to somewhat move the camera and light closer to the scene (b,e), but that for extreme near-field views and light positions, the method fails (a,d). We also empirically found that the lighting augmented neural renderer is more robust to changes in lighting distance.

Finally, our method produces plausible relit results from unstructured photographs. Ideally, our method prefers a good coverage of (the 4D outer-product space of) view and light directions. In particular, one has to be mindful to avoid correlating view and light directions (and thus only sampling a 2D subspace of the 4D view $\times$ light direction space). In our captures, we move the handheld camera with flash light approximately three times faster than the other camera to avoid strong view-light direction correlation.

5 CONCLUSIONS

In this paper we presented a novel deferred neural lighting solution for 360° multi-view relighting from unstructured photographs. Our method is well suited for relighting scenes captured with a dual handheld mobile camera setup. It does not require an accurate estimate of the geometry or of the material properties. Our method combines the advantages of neural textures and deferred lighting in a neural rendering framework. In addition, we introduce a novel refinement augmentation strategy that exploits linearity of light transport to improve generalization of the neural rendering network beyond the training lighting conditions.

For future work we want to explore methods for reducing the number of required photographs, e.g., by embedding better view-interpolation methods and by exploiting appearance similarities in different regions of the scene.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their constructive feedback, Nam *et al.* [2018] for kindly agreeing to help with the comparison with their method, and Xu *et al.* [2018] for sharing their trained network (Figure 10). Pieter Peers was partially supported by NSF grant IIS-1909028. Duan Gao and Kun Xu are supported by the National Natural Science Foundation of China (Project Numbers: 61822204, 61932003, 61521002).

REFERENCES

Martín Abadi, Ashish Agarwal, and et. al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

Jens Ackermann and Michael Goesele. 2015. A Survey of Photometric Stereo Techniques. *Found. Trends. Comput. Graph. Vis.* 9, 3-4 (Nov. 2015), 149–254.

Sai Bi, Z. Xu, K. Sunkavalli, David Kriegman, and Ravi Ramamoorthi. 2020. Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images. In *CVPR*.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured Lumigraph Rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 425–432.

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth Synthesis and Local Warps for Plausible Image-based Navigation. *ACM Trans. Graph.* 32, 3, Article 30 (July 2013).

Anpei Chen, Minye Wu, Yingliang Zhang, Nianyi Li, Jie Lu, Shenghua Gao, and Jingyi Yu. 2018. Deep Surface Light Fields. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 14 (July 2018).

Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N. Kutulakos, and Jingyi Yu. 2020. A Neural Rendering Framework for Free-Viewpoint Relighting. In *CVRP*. 5598–5609.

Robert L. Cook and Kenneth E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (1982), 7–24.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the Reflectance Field of a Human Face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 145–156.

Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. 2018. Single-image SVBRDF Capture with a Rendering-aware Deep Network. *ACM Trans. Graph.* 37, 4, Article 128 (July 2018).

Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. 2019. Flexible SVBRDF Capture with a Multi-Image Deep Network. *Comput. Graph. Forum* 38, 4 (2019).

John Flynn, Michael Broxton, Paul E. Debevec, Matthew DuVall, Graham Fyffe, Ryan S. Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis With Learned Gradient Descent. In *CVPR*. 2367–2376.

Ryo Furukawa, Hiroshi Kawasaki, Katsushi Ikeuchi, and Masao Sakauchi. 2002. Appearance based object modeling using texture database: Acquisition, compression and rendering. In *Rendering Techniques*. 257–265.

Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Trans. Graph.* 38, 4, Article 134 (July 2019).

Rich Geldreich, Matt Pritchard, and John Brooks. 2014. Deferred Lighting and Shading. In *GDC 2014 Presentation*.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. 43–54.

Kaiwen Guo, Peter Lincoln, Philip L. Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Ryan Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul E. Debevec, and Shahram Izadi. 2019. The relightables: volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.* 38, 6, Article 217 (2019).

Tom Haber, Christian Fuchs, Philippe Bekaert, Hans-Peter Seidel, Michael Goesele, and Hendrik P. A. Lensch. 2009. Relighting objects from image collections. In *CVPR*. 627–634.

Daniel Cabrini Hauagge, Scott Wehrwein, Paul Upchurch, Kavita Bala, and Noah Snavely. 2014. Reasoning about Photo Collections using Models of Outdoor Illumination. In *BMVC*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-viewpoint Image-based Rendering. *ACM Trans. Graph.* 37, 6, Article 257 (Dec. 2018).

Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. 2016. Scalable Inside-out Image-based Rendering. *ACM Trans. Graph.* 35, 6, Article 231 (Nov. 2016).

Michael Holroyd, Jason Lawrence, and Todd Zickler. 2010. A Coaxial Optical Scanner for Synchronous Acquisition of 3D Geometry and Surface Reflectance. *ACM Trans. Graph.* 29, 4, Article 99 (July 2010).

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-Image Translation. In *ECCV*. 179–196.

Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C. Sankaranarayanan. 2017. Reflectance capture using univariate sampling of BRDFs. In *ICCV*.

James Imber, Jean-Yves Guillemaut, and Adrian Hilton. 2014. Intrinsic Textures for Relightable Free-Viewpoint Video. In *ECCV*. 392–407.

Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. 2017. Deep View Morphing. In *CVPR*.

Shi Jin, Ruiynag Liu, Yu Ji, Jinwei Ye, and Jingyi Yu. 2018. Learning to Dodge A Bullet: Concyclic View Morphing via Deep Learning. In *ECCV*.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based View Synthesis for Light Field Cameras. *ACM Trans. Graph.* 35, 6, Article 193 (Nov. 2016).

Yoshihiro Kanamori and Yuki Endo. 2018. Relighting Humans: Occlusion-Aware Inverse Rendering for Full-Body Human Images. *ACM Trans. Graph.* 37, 6, Article 270 (Dec. 2018).

Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. 2018. Efficient Reflectance Capture Using an Autoencoder. *ACM Trans. Graph.* 37, 4, Article 127 (July 2018).

Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. 2019. Learning Efficient Illumination Multiplexing for Joint Capture of Reflectance and Shape. *ACM Trans. Graph.* 38, 6, Article 165 (Nov. 2019).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Aldo Laurentini. 2003. The visual hull for understanding shapes from contours: a survey. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, Vol. 1. 25–28.

Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. 2003. Image-Based Reconstruction of Spatial Appearance and Geometric Detail. *ACM Trans. Graph.* 22, 2 (2003).

Anat Levin, Dani Lischinski, and Yair Weiss. 2008. A Closed-Form Solution to Natural Image Matting. *IEEE PAMI* 30, 2 (Feb 2008), 228–242.

Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. 31–42.

Guannan Li, Chenglei Wu, Carsten Stoll, Yebin Liu, Kiran Varanasi, Qionghai Dai, and Christian Theobalt. 2013. Capturing Relightable Human Performances under General Uncontrolled Illumination. *Comput. Graph. Forum* (2013).

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling Surface Appearance from a Single Photograph Using Self-augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4, Article 45 (July 2017).

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018a. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *ECCV*. 74–90.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.* 37, 6, Article 269 (2018).

Miaomiao Liu, Xuming He, and Mathieu Salzmann. 2018. Geometry-Aware Deep Network for Single-Image Novel View Synthesis. In *CVPR*. 4616–4624.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Trans. Graph.* 38, 4, Article 65 (July 2019).

Tom Malzbender, Dan Gelb, and Hans Wolters. 2001. Polynomial Texture Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 519–528.

Abhimitra Meka, Christian Häne, Rohit Pandey, Michael Zollhöfer, Sean Ryan Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul E. Debevec, Christian Theobalt, Julien P. C. Valentin, and Christoph Rhemann. 2019. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Trans. Graph.* 38, 4, Article 77 (2019).

Moustafa Mahmoud Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Kumar Pandey, Noah Snavely, and Ricardo Martin Brualla. 2019. Neural Rerendering in the Wild. In *CVPR*.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.* 38, 4, Article 29 (2019).

Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. 2017. Deep Shading: Convolutional Neural Networks for Screen Space Shading. *Comp. Graph. Forum* (2017).

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H. Kim. 2018. Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography. *ACM Trans. Graph.* 37, 6, Article 267 (Dec. 2018).

Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. 2019. Transformable Bottleneck Networks. In *ICCV*.

Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. 2017. Transformation-Grounded Image Generation Network for Novel 3D View Synthesis. In *CVPR*.

Pieter Peers, Dhruv K. Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. 2009. Compressive Light Transport Sensing. *ACM Trans. Graph.* 28, 1, Article 3 (Feb 2009).

Eric Penner and Li Zhang. 2017. Soft 3D Reconstruction for View Synthesis. *ACM Trans. Graph.* 36, 6, Article 235 (Nov. 2017).

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A. Efros, and George Drettakis. 2019. Multi-view Relighting Using a Geometry-aware Network. *ACM Trans. Graph.* 38, 4, Article 78 (July 2019).

Ravi Ramamoorthi and Pat Hanrahan. 2001. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 497–500.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Trans. Graph.* 34, 4, Article 111 (2015).

Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. 2011. Pocket Reflectometry. *ACM Trans. Graph.* 30, 4, Article 45 (July 2011).

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *CVPR*.

Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *CVPR*. 519–528.

Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Trans. Graph.* 21, 3 (July 2002), 527–536.

Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the Boundaries of View Extrapolation With Multiplane Images. In *CVPR*. 175–184.

Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. 2017. Learning to Synthesize a 4D RGBD Light Field from a Single Image. In *CVPR*. 2262–2270.

Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J. Lim. 2018. Multi-view to Novel view: Synthesizing novel views with Self-Learned Confidence. In *ECCV*.

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi. 2019. Single image portrait relighting. *ACM Trans. Graph.* 38, 4, Article 79 (2019).

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum* 39, 2 (2020), 701–727.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.* 38, 4, Article 66 (2019).

Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. 2009. Dynamic Shape Capture using Multi-View Photometric Stereo. *ACM Trans. Graph.* 28, 5, Article 174 (Dec. 2009).

Michael Weinmann and Reinhard Klein. 2015. Advances in Geometry and Reflectance Acquisition (Course Notes). In *SIGGRAPH Asia 2015 Courses*.

Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. 2000. Surface Light Fields for 3D Photography. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 287–296.

Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. 2016. Recovering Shape and Spatially-varying Surface Reflectance Under Unknown Illumination. *ACM Trans. Graph.* 35, 6, Article 187 (Nov. 2016).

Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. 2019. Deep View Synthesis from Sparse Photometric Images. *ACM Trans. Graph.* 38, 4, Article 76 (July 2019).

Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF Sampling for Two-shot Near-field Reflectance Acquisition. *ACM Trans. Graph.* 35, 6, Article 188 (Nov. 2016).

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep Image-based Relighting from Optimal Sparse Samples. *ACM Trans. Graph.* 37, 4, Article 126 (July 2018).

Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. 2016. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*. 1696–1704.

Jimei Yang, Scott Reed, Ming-Hsuan Yang, and Honglak Lee. 2015. Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis. In *NIPS*. 1099–1107.

Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2018. Single Photograph Surface Appearance Modeling with Self-Augmented CNNs and Inexact Supervision. *Comput. Graph. Forum* 37, 7 (Oct 2018).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis Using Multiplane Images. *ACM Trans. Graph.* 37, 4, Article 65 (July 2018).

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016b. View Synthesis by Appearance Flow. In *ECCV*.

Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. 2016a. Sparse-as-possible SVBRDF Acquisition. *ACM Trans. Graph.* 35, 6, Article 189 (Nov. 2016).

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*.