Recovering Real-World Reflectance Properties and Shading From HDR Imagery

Bjoern Haefner[1,2] Simon Green[2] Alan Oursland[2] Daniel Andersen[2]
Michael Goesele[2] Daniel Cremers[1] Richard Newcombe[2] Thomas Whelan[2]
[1]Technical University of Munich [2]Facebook Reality Labs Research
{bjoern.haefner, cremers}@tum.de,
{simongreen, ours, andersed, goesele, newcombe, twhelan}@fb.com

Abstract

We propose a method to estimate the bidirectional reflectance distribution function (BRDF) and shading of complete scenes under static illumination given the 3D scene geometry and a corresponding high dynamic range (HDR) video. By splitting the BRDF into its diffuse and non-diffuse parts we solve the estimation of each component separately. For the diffuse component, we sample the incident illumination at each point in the scene using Monte Carlo ray tracing, allowing us to factor the captured surface color into albedo and shading. We then use a novel ray tracing-based optimization strategy to estimate the non-diffuse parameters of the BRDF. In a variety of experiments, we demonstrate that our method efficiently generates realistic copies of the observed scenes.

1. Introduction

Recovering a faithful copy of our world is of fundamental importance for virtual, augmented and mixed reality (VR, AR, MR). VR devices immerse the user into a virtual world to fulfill certain tasks, e.g. medical, educational or gaming purposes. They rely on a representation of the scene in terms of surface geometry, material properties and lighting. Since hand-crafting such virtual world models is tedious, there is an increasing demand for methods that can automatically reconstruct real world environments. Yet, their practical value critically depends on the realism of the virtual world. In MR and AR, faithful scene representations are required to render virtual objects that have the correct physical interactions and visual appearance with respect to their surroundings. While the reconstruction of surface geometry is quite mature, the estimation of reflectance and lighting of a scene remains a difficult open challenge – in particular, if we want to estimate these properties straight from an input video. This work brings the virtual and real world closer together enabling users to immerse in a *realistic virtual reality* and experience believable augmentations.

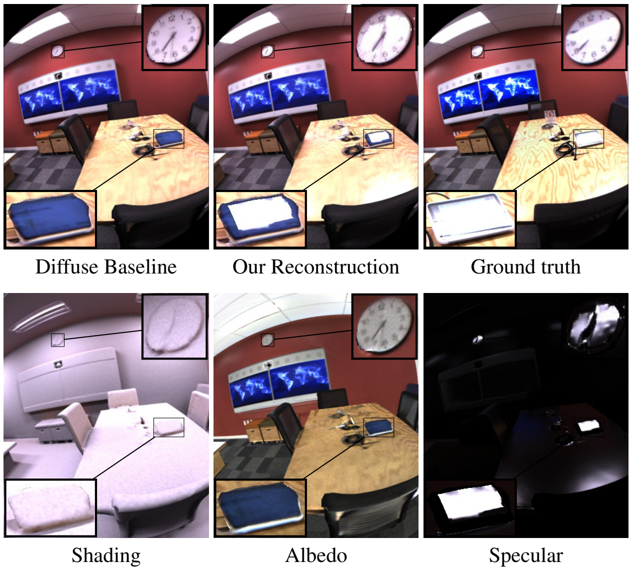


Figure 1. Reconstruction results: Given an input video and a geometric reconstruction of the scene, we deduce the scene's shading, albedo and specular properties, thereby allowing for a more faithful reconstruction. The insets show details of two specular objects.

Given a comprehensive HDR video of an environment and its corresponding reconstructed 3D mesh, we claim three novel contributions:

- An efficient method to leverage HDR textures for estimating albedo and shading per surface element.
- A procedure to calculate ideal target frames for each object in the scene within the estimation process.
- A method to estimate the non-diffuse BRDF using grid search with nested least-squares optimization.

On a broad range of real-world datasets, we demonstrate that this enables faithful reconstructions, plausible scene relighting and visually accurate rendering of virtual objects that can take the surrounding scene appearance and geometry into account.

2. Background and related work

In the following we recall the rendering equation [18] and discuss efforts to invert it in order to recover realistic models of the observed world.

2.1. The rendering equation

The rendering equation [18] is a useful and popular tool to render images given the scenes properties of material, illumination and geometry. It models the light transport as:

$$L_{\mathrm{o}}(\boldsymbol{x},\omega_{\mathrm{o}})=L_{\mathrm{e}}(\boldsymbol{x},\omega_{\mathrm{o}})+\int_{\mathcal{H}^2} f_{\mathrm{r}}(\boldsymbol{x},\omega,\omega_{\mathrm{o}})L(\boldsymbol{x},\omega)\langle\omega,\boldsymbol{n}\rangle\mathrm{d}\omega \quad (1)$$

where the L_{o} is the observed radiance at $\boldsymbol{x} \in \mathbb{R}^3$ in direction $\omega_{\mathrm{o}} \in \mathbb{S}^2$, with $\mathbb{S}^2$ being the 3D unit sphere. $L_{\mathrm{e}}(\boldsymbol{x},\omega_{\mathrm{o}})$ describes the amount of light emitted at $\boldsymbol{x}$ in direction ω_{o} by a light source. The integral over the hemisphere $\mathcal{H}^2$ oriented by the surface normal $\boldsymbol{n} \in \mathbb{S}^2$ positioned at $\boldsymbol{x}$, integrates along all incident directions ω. The integrand describes the interaction between material, light and geometry, where the BRDF f_{r} models the reflectance properties of a variety of materials. The radiance $L(\boldsymbol{x},\omega)$ describes the amount of incoming light at $\boldsymbol{x}$ from direction ω. The geometric term $\langle\omega,\boldsymbol{n}\rangle$ models the spread of incident illumination over the surface at a given angle, where $\langle\cdot,\cdot\rangle : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ describes the dot product. Evaluating Eq. (1) can result in high quality renderings close to real-world images [18, 37], providing the relation between a captured image and its scene. To this end we identify for each pixel $\boldsymbol{p} \in \Omega \subset \mathbb{R}^2$ of the image $I : \Omega \to \mathbb{R}^3$, the conjugate 3D-point $\boldsymbol{x}$. And ω_{o} is the normalized vector pointing from $\boldsymbol{x}$ to $\boldsymbol{p}$, $I(\boldsymbol{p}) = L_{\mathrm{o}}(\boldsymbol{x},\omega_{\mathrm{o}})$.

2.2. Inverting the rendering equation

Inferring camera and scene properties by inverting the rendering equation in order to obtain suitable models of the real world has a long-standing history and is called inverse rendering [35]. We now discuss the most related work that tackles the challenging task of material estimation, but refer to [20] for a comprehensive survey on inverse rendering.

Deep Learning [4, 5, 9, 24, 25, 26, 30, 43, 45, 54] approaches train a network in an (un-)supervised manner and demonstrate impressive results in the context of photorealistic scene reconstruction. Yet, these techniques applied in the single image domain, are concerned with single object reconstruction, and/or have an implicit scene representation. This makes it difficult to be compatible with conventional computer graphics assets used for lighting and physics interactions in full 3D room-scale real-world reconstructions, thus limiting the applicability of these approaches to AR/VR/MR applications.

Multi-Shot [1, 3, 4, 9, 11, 13, 21, 22, 26, 23, 27, 29, 30, 31, 33, 40, 42, 52, 53] techniques recover material effects using multiple images taken from the same or different viewpoints. While more observations constrain the resulting optimization problem better, additional images need to be captured and the computational burden can limit inference in terms of memory and runtime. Thus, it is always desirable to use as few images as possible, while still constraining the search space of possible solutions enough to find reasonable estimates. Additionally, many of these approaches have at best a piece-wise constant material per object if no active lighting is used.

Active Lighting [1, 9, 11, 13, 16, 15, 21, 22, 31, 36, 38, 42, 53] frameworks estimate reflectance properties similar to multi-shot techniques, but additionally require different (calibrated) illumination for each image. This limits the practicability of these approaches as a light source has to be actively controlled. It is known that these approaches are well-posed in the Lambertian setting under general illumination [6] and a point-wise solution of the albedo can be found as it is much more constrained. Considering view-dependent material effects adds additional complexity to the problem and even if illumination and geometry is known, recovering non-diffuse reflectance is an open challenge and additional assumptions have to be made [14].

HDR Imagery [1, 11, 21, 22, 29, 52, 53] shows great usage in photometric approaches as they tend to relate scene properties to linear radiance data instead of non-linearly mapped pixel intensities [12] – a relation which, if violated due to no camera calibration, can result in undesired deterioration [12, 19]. Interestingly, despite its potential and desirable properties the literature applying HDR data in the context of photorealistic reconstruction of room-sized environments is fairly sparse [29, 52]. This might be due to the different orders of magnitude involved when using HDR data – an effect non-existent with 8-bit images as higher radiance values are usually clamped to 255. This can cause standard algorithms, like running average of pixel intensities to not work as expected.

In contrast to the above approaches, the method presented here does not rely on large amounts of diverse training data, nor on active lighting and works in complete room-sized 3D environments. We effectively incorporate the advantages of HDR imagery and a tailored ray tracing framework to recover the BRDF for every object in the scene. More specifically, we can recover a spatially varying albedo, and present a principled way to leverage HDR video for automated target frame selection which allows us to estimate non-diffuse material effects from a single view per object. To the best of our knowledge this is the first work utilizing HDR data with consistent full 3D room-scale reconstructions, which is able to recover BRDF parameters of every object in the scene using a single automatically computed target frame. In the context of AR/VR/MR, this enables the faithful recovery of large-scale scenes that support conventional physical as well as light interaction between real and virtual objects.

3. Recovering complex reflectance and shading

Given a mesh-based 3D reconstruction of the scene geometry, an HDR RGB sequence of frames covering that geometry and their corresponding poses, we first reconstruct and estimate the lit diffuse HDR texture (Section 3.2). This then builds the foundation for the albedo and shading estimation using only the textured geometry (Section 3.3), and, given an object segmentation, the estimation of the specular material parameters per object (Section 3.4). See Algorithm 1 for an overview of our proposed framework. Note that our input assumptions differ only in the HDR data compared to other approaches like [3], allowing us to cover the dynamic range of the scene from the darkest to the brightest areas. We follow [12] to transform the captured data to floating point linear units directly proportional to the incoming radiance and discuss in Section 3.2 and 3.4.1 arising issues and how to effectively leverage that to our advantage.

Algorithm 1 Overview of our proposed algorithm

```
Input: HDR data, poses, geometry, object segmentation
Output: $\tilde{\boldsymbol{\rho}}$, $\{\varphi^i, \psi^i\}_{i=1,\dots,M}$ for all $M$ objects in the scene
    Calculate lit diffuse HDR texture (Sec. 3.2):
    $L_\mathrm{d}$ = runningMedian(HDR data, poses, geometry)
    Calculate shading $S$ and albedo $\tilde{\boldsymbol{\rho}}$ (Sec. 3.3):
    $S$ = calcShading(geometry, $L_\mathrm{d}$)
    $\tilde{\boldsymbol{\rho}} = \frac{L_\mathrm{d}}{S}$
    For each object: Target frame calculation and roughness $\varphi^i$ and specular $\psi^i$ estimation (Sec. 3.4):
    TFs = calcTargetFrames(HDR data, poses, geometry, object segmentation)
    for each object $i$ in the scene do
        TF = TFs[i] ($i$-th object's target frame)
        $\varphi^i$, $\psi^i$ = estimateNondiffuse(TF, geometry, $L_\mathrm{d}$)
```

3.1. Microfacet BRDF Model

We restrict our focus to isotropic, dielectric (non-metallic), and opaque (not translucent/transparent) objects only. A desirable property for a BRDF is an additive separation into its diffuse and non-diffuse component, as it allows splitting the problem of BRDF parameter estimation into two separate, easier to solve problems as we will describe later. We will thus use a dichromatic BRDF [44],

$$f_\mathrm{r}\left(\boldsymbol{x},\omega,\omega_\mathrm{o}\right) = f_\mathrm{d}\left(\boldsymbol{x}\right) + f_\mathrm{nd}\left(\boldsymbol{x},\omega,\omega_\mathrm{o}\right) \tag{2}$$

and identify the diffuse part as $f_\mathrm{d}\left(\boldsymbol{x}\right) = \frac{\boldsymbol{\rho}(\boldsymbol{x})}{\pi} =: \tilde{\boldsymbol{\rho}}\left(\boldsymbol{x}\right)$, and call $\tilde{\boldsymbol{\rho}}$ the *(scaled) albedo*, where $\boldsymbol{\rho} : \Sigma \to [0,1]^3$, given a reconstructed surface $\Sigma \subset \mathbb{R}^3$. The non-diffuse component is described using the Torrance-Sparrow microfacet model [10, 49] with a GGX distribution [46, 50, 51] and Schlick's Fresnel approximation [41] (dropping the $\boldsymbol{x}, \omega, \omega_\mathrm{o}$ dependencies for brevity),

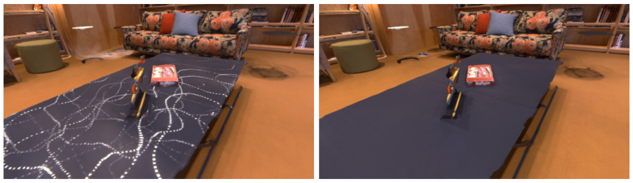

Running Mean Running Approximated Median

Figure 2. The mean textures (left) suffer from occluding edge bleeding and baked in specularities, our approximated median (right) is able to estimate textures without such artifacts.

$$\begin{aligned} f_\mathrm{nd}\left(\varphi,\psi\right) =& G\left(\varphi\right) D\left(\varphi\right) F\left(\psi\right), & (3)\\ G\left(\varphi\right) =& G_1\left(\left\langle \boldsymbol{n},\omega\right\rangle, \tilde{\varphi}\right) \cdot G_1\left(\left\langle \boldsymbol{n},\omega_\mathrm{o}\right\rangle, \tilde{\varphi}\right), & (4)\\ D\left(\varphi\right) =& \frac{\hat{\varphi}^2}{\pi\left(1+\left(\hat{\varphi}^2-1\right)\left\langle \boldsymbol{n},h\right\rangle^2\right)^2}, & (5)\\ F\left(\psi\right) =& \tilde{\psi} + \left(1-\tilde{\psi}\right)\left(1-\left\langle \omega,h\right\rangle\right)^5, & (6) \end{aligned}$$

with $G_1\left(x,y\right) = \left(x + \sqrt{x^2+y^2-x^2y^2}\right)^{-1}$, the half vector $h = \frac{\omega+\omega_\mathrm{o}}{\|\omega+\omega_\mathrm{o}\|}$, and the nondiffuse parameters *roughness* $\varphi : \Sigma \to [0,1]$, and *specular* $\psi : \Sigma \to [0,1]$. Following [7], we apply three reparameterisations to increase robustness: $\tilde{\varphi} = (\frac{\varphi}{2} + \frac{1}{2})^2$ to have a more perceptually linear change in roughness, $\hat{\varphi} = \max\left(0.001,\varphi\right)$ for numerical stability, and $\tilde{\psi} = 0.08\psi$ causing the refractive index to cover most common materials. Plugging (2) into (1), assuming non-emissivity ($L_\mathrm{e} \equiv 0$) and splitting the integral, we get

$$\begin{aligned} L_\mathrm{o}\left(\boldsymbol{x},\omega_\mathrm{o}\right) =& L_\mathrm{d}\left(\boldsymbol{x}\right) + L_\mathrm{nd}\left(\boldsymbol{x},\omega_\mathrm{o}\right), & (7)\\ L_\mathrm{d}\left(\boldsymbol{x}\right) :=& f_\mathrm{d}\left(\boldsymbol{x};\boldsymbol{\rho}\right)\int_{\mathcal{H}^2} L\left(\boldsymbol{x},\omega\right)\left\langle \omega,\boldsymbol{n}\right\rangle \mathrm{d}\omega, & (8)\\ L_\mathrm{nd}\left(\boldsymbol{x},\omega_\mathrm{o}\right) :=& \int_{\mathcal{H}^2} f_\mathrm{nd}\left(\boldsymbol{x},\omega,\omega_\mathrm{o};\varphi,\psi\right) L\left(\boldsymbol{x},\omega\right)\left\langle \omega,\boldsymbol{n}\right\rangle \mathrm{d}\omega. & (9) \end{aligned}$$

3.2. Lit diffuse HDR texture estimation

We estimate the *lit diffuse HDR texture* by projecting the video frames onto the surface geometry. Using low dynamic range 8-bit data, weighted averaging [8, 32] typically yields reasonable results as outliers are smoothed out. However, this is not the case with HDR data due to its large range of values, resulting in a number of visual artifacts caused by two main phenomena: errors in the geometry and bright lights along with specular reflections of those, see Fig. 2 left. One popular approach to diminish these artifacts is to calculate the median rather than a running mean [39]. However, this is extremely expensive since it requires storing all RGB values. To overcome this we estimate an approximation of the median of each color channel using the P-Square algorithm [17][1]. Figure 2 shows a comparison between the running mean and our running approximated median. Despite errors in the reconstruction, the floor is no longer corrupted and specular reflections on the table have been removed. Mathematically speaking, the BRDF inscribed in the texture should have no view-dependent effects and can thus be assumed to represent the albedo. Nevertheless, one should not identify the median texture with the albedo itself as it still contains global light transport and geometric information. Thus, we assume that the median texture can be identified as the diffuse radiance, L_{d} and we call it the *lit diffuse HDR texture*.

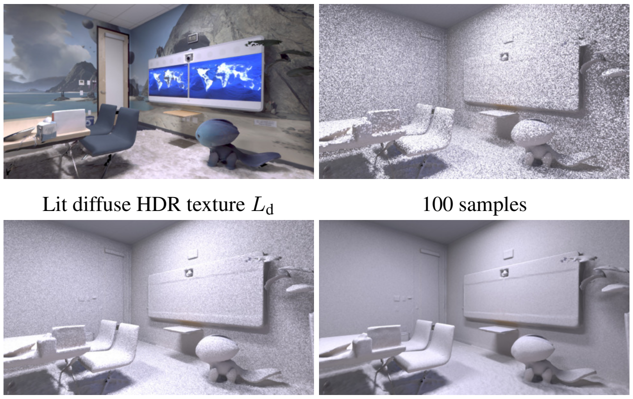
3.3. Albedo and shading estimation

We are now going to effectively leverage the information that the HDR texture's intensity is proportional to the true radiance, which would not be possible with textures where intensities, especially at light sources, are truncated to 8-bits. This allows us to estimate the captured shading S at each surface point $\boldsymbol{x} \in \Sigma$ of the scene,

$$S\left(\boldsymbol{x}\right) := \int_{\mathcal{H}^2} L\left(\boldsymbol{x}, \omega\right) \langle \omega, \boldsymbol{n} \rangle \, \mathrm{d}\omega. \tag{10}$$

The shading describes the sum of the radiance $L\left(\boldsymbol{x}, \omega\right)$ gathered from the scene weighted by the geometric scale factor $\langle \omega, \boldsymbol{n} \rangle$. We estimate the shading S via Monte-Carlo ray tracing, a stochastic approach to estimate complex integrals such as Eq. (10). We cast rays at each scene's surface point $\boldsymbol{x} \in \Sigma$ on the hemisphere $\mathcal{H}^2$, where the chosen ray directions ω follow a distribution accounting for the scalar product in Eq. (10) (cosine weighted) [37]. For each cast ray $\left(\boldsymbol{x}, \omega\right)$ we read the lit diffuse HDR texture at the closest hit point $\tilde{\boldsymbol{x}}$ and interpret it as the incident radiance, $L\left(\boldsymbol{x}, \omega\right) = L_{\mathrm{d}}\left(\tilde{\boldsymbol{x}}\right)$. Summing up all cosine weighted samples of incident radiance gives an estimate for the shading S for each surface point $\boldsymbol{x}$. One can interpret this procedure as sampling each surface point's environment map. Note that the captured lit diffuse HDR texture already includes the effects of global light transport in the diffuse scene [52], thus we can perform the proposed shading estimation in parallel for all surface points $\boldsymbol{x} \in \Sigma$ independently. Finally, as the captured lit diffuse HDR texture is the product of the scaled albedo and the shading, see Eq. (8), we can recover the albedo by dividing the captured lit diffuse HDR texture by the estimated shading.

Fig. 3 shows shading estimates for different numbers of ray samples and how increasing sample size de-noises the result. Our approach to recover shading and albedo does not account for emissive radiance L_{e}, although the lit diffuse HDR textures inherently carries that information. Nevertheless, we do not think of this as a major disadvantage, as for emissive objects, the impact of the intrinsic radiance (what we see when looking at a light source which is turned off) is negligible compared to its emissive radiance.


Lit diffuse HDR texture L_{d}

100 samples

1000 samples

10000 samples

Figure 3. Estimated shading S for different numbers of ray samples. Note how more samples remove noise from the shading.

3.4. Specular appearance estimation

Given an estimate of the lit diffuse scene $L_{\mathrm{d}}\left(\boldsymbol{x}\right)$ at each surface point, we can estimate the non-diffuse BRDF parameters ψ and φ per object. We assume that for all M objects in the scene, each object's view-dependent effects can be described with two parameters, $\left\{\left(\psi^i, \varphi^i\right)\right\}_{i=1,\ldots,M}$, resulting in two constant, non-diffuse BRDF parameters per object. We first discuss how we automatically select an individual target frame per object given an object segmentation before utilizing these in the proposed optimization scheme.

3.4.1 Target frames

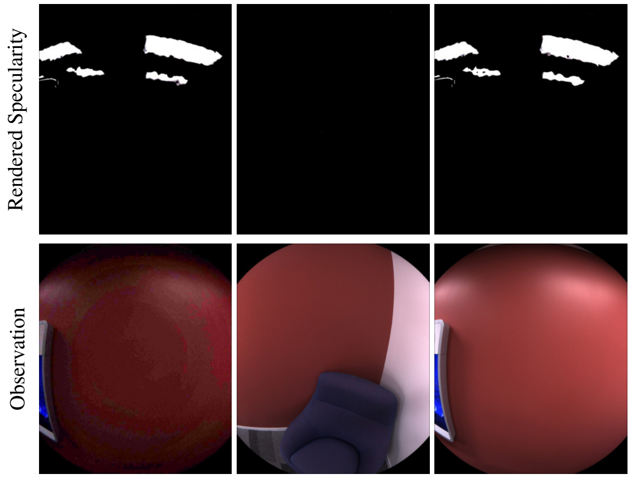
Path tracing a single image is expensive and time-consuming, which is why we would like to use as few images for inference as possible. Additionally, the so called target frames (TF) used to estimate each object's non-diffuse material parameters should have two attributes:

- $\mathcal{A}_1$, high chance of specular highlights caused by direct illumination, and
- $\mathcal{A}_2$ the captured observation from the HDR video consists mostly of valid pixels, i.e., the RGB values are not over- or under-saturated[2].

These requirements in combination with the assumption that a single object's specular appearance can be described with two parameters allows us to use *only one* TF per object. In order to find TFs fulfilling $\mathcal{A}_1$, we assume the object of interest is a perfect mirror and we render only the pixels where light sources can be seen in the mirrored surface.

[1] In the interest of brevity we refer to the original paper for a full description of the algorithm and its performance relative to an exact median.

[2] Over- or under-saturated observations do not depend linearly on the incoming radiance [12], hence we omit them to avoid corrupting the result.



$\mathcal{A}_1$	✓	✗ (no highlights)	✓
$\mathcal{A}_2$	✗ (under-saturated)	✓	✓
TF	✗	✗	✓

Figure 4. Example of good and bad target frame (TF) candidates for the object "Red Wall" based on the attributes $\mathcal{A}_1$ and $\mathcal{A}_2$. While for the first two columns, either the observation is under-saturated (intensity increased by a factor of 10 for visualization purposes) or there are no specular highlights on the object, the third column shows a TF satisfying both $\mathcal{A}_1$ and $\mathcal{A}_2$.

Note that this is the only step in our framework that requires information about position of emitting light sources. Concerning $\mathcal{A}_2$, the HDR capture cycles through three different exposures in subsequent frames. This leads to three different exposures at roughly the same viewpoint, allowing us to find at least one frame with enough valid pixels. Example frames and their attributes $\mathcal{A}_1$ and $\mathcal{A}_2$ are visualized in Fig. 4. We iterate through the video and for each of it's objects set the TF as the frame with most pixels in $\mathcal{A}_1 \cap \mathcal{A}_2$.

3.4.2 Optimization

Given the i-th object's target frame I^i we describe it as the composition of its diffuse and non-diffuse component, I^i_{d} and I^i_{nd} respectively,

$$I^i(\boldsymbol{p}) = I^i_{\text{d}}(\boldsymbol{p}) + I^i_{\text{nd}}\left(\boldsymbol{p};\varphi^i,\psi^i\right). \tag{11}$$

We assume I^i (the observation) and I^i_{d} (rendered image using the lit diffuse HDR texture) to be given so that the only varying quantity is I^i_{nd}. Due to view-dependent appearance effects, full evaluation, i.e., dense sampling of ω of the integral in Eq. (9) is challenging. We therefore follow a multi importance sampling strategy [37]. For further technical details see the supplementary material. We assume that single bounce ray tracing is enough for a reasonable approximation of the scene [52] keeping computational expense practical. We have a good estimation of the lit diffuse scene thanks to the HDR median textures L_{d}. When adding view-dependent effects such as reflections, inter-reflections start to have impact on the final result. Nevertheless, a specular lobe illuminating the scene is assumed to be negligible compared to an emissive light source when integrating over the whole hemisphere, as our target frames are chosen such that specular reflections are mainly caused by direct illumination ($\mathcal{A}_1$). Even in the presence of mirror like objects our target frames were not corrupted enough with indirect illumination that this would cause the system to fail. In order to determine the non-diffuse properties of the BRDF we can now formulate an optimization problem in $\mathcal{X}^i := (\varphi^i, \psi^i)$ per object i, i.e., we want to solve for $i = 1, \ldots, M$,

$$\min_{\mathcal{X}^i\in[0,1]^2} \mathcal{L}(\mathcal{X}^i) := \sum_{\boldsymbol{p}\in\Omega^i} \left\| r\left(\boldsymbol{p};\mathcal{X}^i\right)\right\|_2^2 . \tag{12}$$

$\|\cdot\|_2$ is the L_2-norm and r a point-wise RGB-color residual at each pixel $\boldsymbol{p}$ in the image domain $\Omega^i \subset \Omega$ showing only the i-th object,

$$r\left(\boldsymbol{p};\mathcal{X}^i\right) = I^i(\boldsymbol{p}) - \left(I^i_{\text{d}}(\boldsymbol{p}) + \mathcal{I}^i_{\text{nd}}\left(\boldsymbol{p};\mathcal{X}^i\right)\right). \tag{13}$$

Note that due to the single bounce assumption, the M optimization problems in (12) are disjoint, which enables solving each problem separately and in parallel.

Optimization problems like Eq. (12) are difficult to solve due to the non-convexity in the roughness parameter φ^i, cp. Eqs. (4) and (5). We now present a simple and fast numerical scheme that can tackle the inherent complexity by exploiting the closed parameter domain $[0,1]^2$ of $\mathcal{X}^i$ and the fact that the BRDF f_{nd} depends only linearly on the specular parameter ψ^i, cp. Eq. (6). We perform a two-level grid search approach (in φ^i) with nested least squares optimization (in ψ^i). That is, at the l-th level we set the roughness $\varphi^i = \varphi^i_{l_k}$ from a discrete set of sample points equidistantly spread across an interval $[a_l, b_l]$,

$$\varphi^i_{l_k} \in \{a_l + \frac{k\cdot(b_l - a_l)}{K-1} \mid k = 0,\ldots,K-1\} \tag{14}$$

For each $\varphi^i_{l_k}$ we calculate the best specular value $\psi^i_{l_k}$ by solving the linear least squares problem in Eq. (12) in closed form. For the resulting K tuples $\{\mathcal{X}^i_{l_k}\}_{k=0,\ldots,K-1}$ at the l-th level we evaluate $\mathcal{L}(\mathcal{X}^i_{l_k})$ and set the minimizer of the l-th level as the one with lowest cost. We choose $K = 11$, as we found this gives a dense enough sampling of the interval $[a_l, b_l]$ $\forall l$. At the first level we set $a_0 = 0$, $b_0 = 1$, while the second level's interval is initialized with the direct left and right neighbours of the minimizer's roughness value, or the roughness value itself in case it lies on the boundary of the sampling interval. Note that this approach always terminates after $K\cdot$"number of levels"$= 22$ iterations, but is not guaranteed to find a global minimizer – a challenging task in non-convex optimization. In our evaluation we did not observe any failed results that were undoubtedly caused by an unsuccessful optimization.

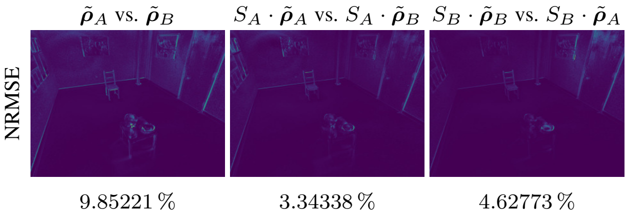


Figure 5. Error maps and numbers of normalized RMSE (NRMSE) of our approach to estimate albedo and shading on two differently illuminated scans A and B. We compare the two estimated albedos $\tilde{\boldsymbol{\rho}}_A$ and $\tilde{\boldsymbol{\rho}}_B$, and how well the ground truth ($S_A \cdot \tilde{\boldsymbol{\rho}}_A$ and $S_B \cdot \tilde{\boldsymbol{\rho}}_B$) can be predicted with the other scan's albedo ($S_A \cdot \tilde{\boldsymbol{\rho}}_B$ and $S_B \cdot \tilde{\boldsymbol{\rho}}_A$). See Figure 6 for the images used to calculate the shown error maps.

4. Experiments

Given each surface point's albedo and shading, as well as each object's specular appearance, we can now quantitatively and qualitatively evaluate the effectiveness of our proposed approach. We use the Replica dataset [48] for the evaluation as this provides appropriate input data: reconstructed meshes of the complete scene, HDR video (provided by the authors of [48]), per frame camera poses, and semantic object instance information.

For quantitative validation of the albedo and shading estimation we use a dataset captured by ourselves with control over illumination and the objects in the scene, see the supplementary material for details on the capturing process. The room has in total four globe lights and three LED panels as light sources, which differ in wavelength and emission. The two scans differ in their respective lighting: For the first scan, all four globe lights and one LED panel were turned on (we call this *Scan/Reconstruction* A). For the second scan, only two LED panels (different from the one in *Scan* A) were turned on (we call this *Scan/Reconstruction* B). Note that we calculate the set of lit diffuse HDR textures (Section 3.2) a priori for each dataset, which runs on the GPU at ≈8–9Hz for RGB images of resolution 1224×1024. All experiments are carried out on a machine with an Intel Xeon 3.70GHz and an NVIDIA GeForce RTX 2080. *We encourage the reader to view our supplementary material for further results.*

4.1. Albedo and shading validation

We use *Reconstructions* A and B as well as scenes from the Replica dataset [48] to evaluate the albedo and shading described in Section 3.3. Using NVidia's OptiX engine [34], we cast 10000 rays per texel from each corresponding surface element to get a de-noised estimate of the shading and albedo. This process takes ∼10min.

Quantitative evaluation is carried out on the *Reconstructions* A and B. The reconstructed albedos should ideally be equal as lighting cues are explained by the shading S. Figure 5 shows the normalized RMSE (NRMSE) for the whole scene verifying that there is only little, i.e. less than 10% difference between the two albedos. Additionally, a numerical evaluation between the ground truth and predicted reconstructions is carried out. To this end, we compare $\tilde{\boldsymbol{\rho}}_A \cdot S_A$ vs. $\tilde{\boldsymbol{\rho}}_B \cdot S_A$ to see how well reconstruction A can be predicted, while $\tilde{\boldsymbol{\rho}}_B \cdot S_B$ vs. $\tilde{\boldsymbol{\rho}}_A \cdot S_B$ validates the prediction of reconstruction B. An error well below 5% for both tests shows that we can faithfully modify diffuse scenes with novel lighting conditions. Figure 6 shows the estimated albedos, shadings, ground truth and their predictions. While overall both albedo estimates are visually almost identical, few artifacts are visible and show how our system performs under violated assumptions of 1) remaining view-dependent effects in the lit diffuse HDR texture (e.g. on the door), and 2) inaccuracies in the reconstructed geometry (e.g. the objects on the table). Nevertheless, as numerically and quali-

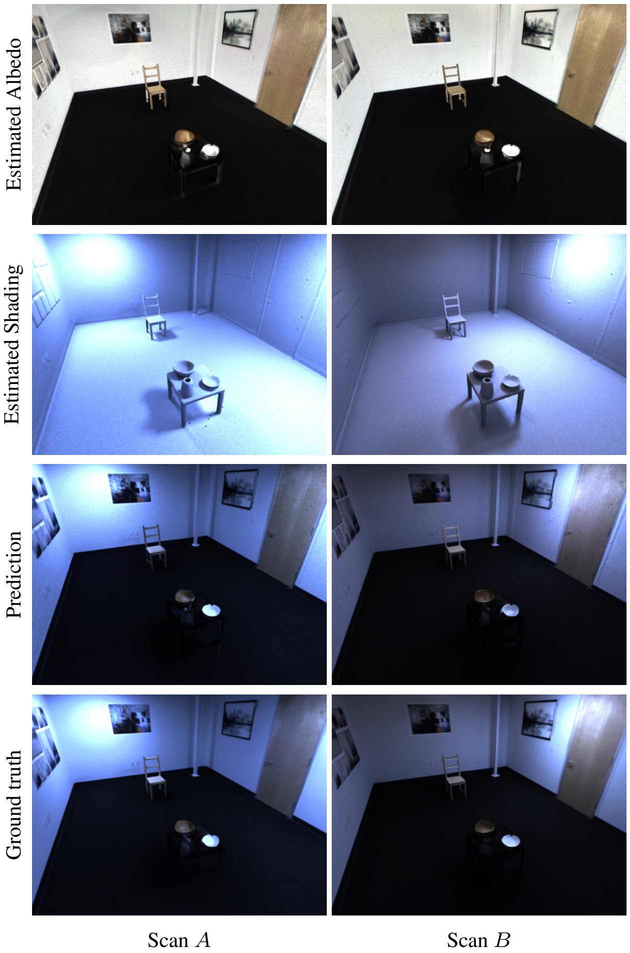


Figure 6. Numerical evaluation of our approach to estimate albedo and shading on two differently illuminated scans A and B. As can be seen visually, illumination information is nicely explained in the estimated shading, while both albedos look almost identical and the predicted scene is close to ground truth.

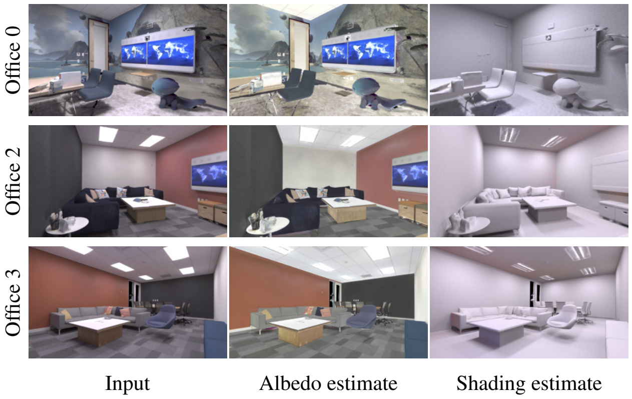
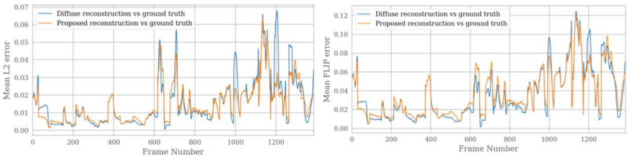


Figure 7. We deploy our albedo and shading estimation on challenging real-world “Office” data sets of the Replica data set [48] and are able to estimate per-texel albedo and shading information, using the reconstructed mesh and lit diffuse HDR texture only. More results can be found the in the supplementary material.



Mean L_2 error

Mean FLIP error

Figure 8. Numerical comparison on Office 1 [48] between a purely diffuse reconstruction with the ground truth (blue line) and the proposed reconstruction with the ground truth (orange). The left shows the numerical mean L_2 metric, while the right visualises the perceptual FLIP [2] metric. More results can be found in the supplementary material.

tatively shown, errors in our albedo estimation are still tolerable to plausibly relight diffuse scenes, i.e. errors between the two albedo estimates are easier to detect than errors between predicted relighting and ground truth.

Qualitative evaluation is carried out on the real-world Replica dataset [48] and can be seen in Figure 7. When our assumptions are met, we can recover an albedo estimate free of illumination effects, as these are contained in the corresponding shading estimate. Furthermore, we are able to tackle the challenging task of removing cast shadows of objects, e.g., chairs, sofas and tables. Note that in Office 0 the table under the display has a stand right below it on the floor which can be mistaken as a cast shadow in the albedo estimate, but the corresponding shading estimate shows it has actually been successfully removed.

4.2. Specular appearance estimation validation

For quantitative and qualitative comparison, we deploy our approach described in Section 3.4 on the Replica dataset [48]; casting 200 rays per each pixel’s corresponding surface element using OptiX [34]. The dataset consists of $\approx$50–150 objects per scene; each of different size, geometry and material. Estimating an object’s non-diffuse BRDF parameters takes $\approx$238sec on the GPU.

Quantitative evaluation is concerned with how much a reconstruction improves compared to the *diffuse baseline*, i.e. a reconstruction using the lit diffuse HDR textures. We infer non-diffuse material parameters from a single image per object. More specifically, to validate consistency across different views we test our predictions against unseen viewpoints of the ground truth observation, and compare this to the diffuse baseline. To this end we use two different error metrics, the numerical L_2-loss, as well as the recently introduced perceptual FLIP [2] evaluator. FLIP has a particular focus on the differences between rendered images and corresponding ground truths via approximating the difference perceived by humans when alternating between two images. Figure 8 shows the per frame numerical mean L_2 metric (left), and the perceptual mean FLIP [2] metric (right) for a video sequence of Office 1 [48] containing 1389 frames, where 1363 frames are novel viewpoints and only 26 frames were used as target frames. Both graphs show that on average the error decreases when incorporating the proposed view-dependent BRDF estimates. Note that, besides only small differences between the orange and blue graph (as specular highlights are only sparsely distributed across an image, if they appear at all), the improvements (“orange$<$blue”) are of much larger magnitude than the deterioration (“blue$<$orange”). That means that if our proposed rendering degrades the ground truth more than the diffuse baseline, it is only slightly worse, while our proposed rendering considerably improves realism compared to the diffuse baseline.

Qualitative evaluation and comparison to related work is carried out over multiple real-world datasets of [48], see Figure 9. The state-of-the-art approach closest related to ours is [3], which is a full path tracing (2 bounces) approach to estimate the scene’s material properties. We chose the hyper-parameters as recommended by the authors using 1 sample to estimate the image and 511 for the derivative and ran [3] until convergence which took 12$-$24hrs, depending on the data set. A side-by-side comparison between the diffuse, the state-of-the-art [3] and the proposed reconstruction along with the ground truth and the corresponding error maps shows the superiority of our approach. While the overall trend of estimated material parameters of [3] seems correct, Monte Carlo noise is dominating the resulting reconstruction which heavily deteriorates the rendered images. Our method can successfully reconstruct subtle specular effects such as the specular lobe on the wall in Office 0. It also models stronger reflections, e.g., the TV screen in Office 4 and even mirror like reflections, see the glass window in Office 3. Inaccuracies in geometry can affect the result (Office 1, largest deterioration according to Fig. 8), although the tablet glass screen seems to be well estimated.

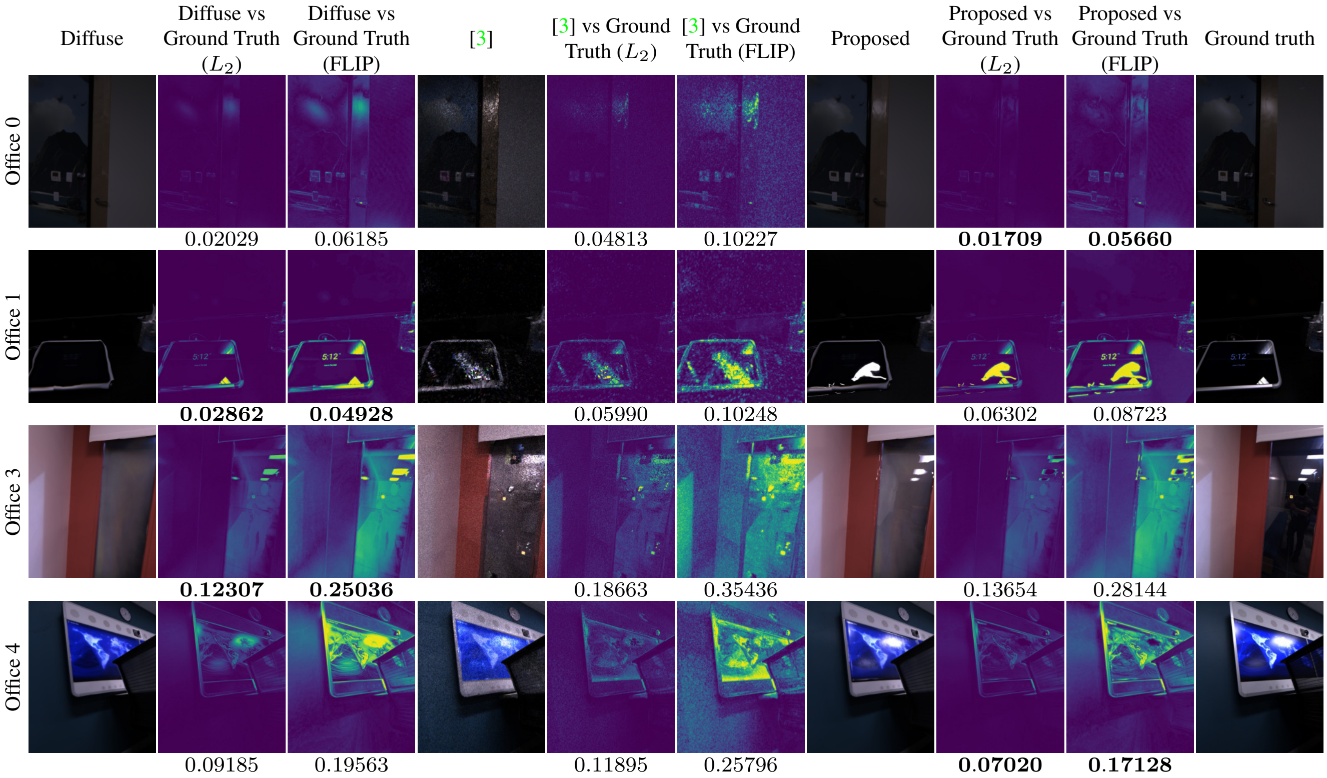


Figure 9. Side-by-side comparisons between the diffuse baseline, a path tracing approach [3] and the proposed reconstruction along with the ground truth and the corresponding L_2 errors and FLIP evaluator [2]. More results can be found in the supplementary material.

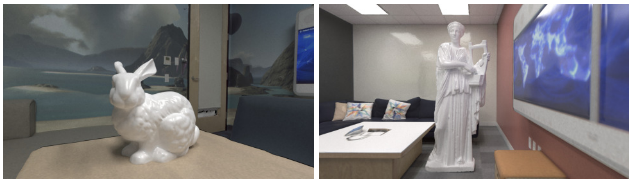


Figure 10. Complete synthetic relighting of different data sets (Office 0, Office 2 of [48]) with virtually placed objects [47, 28]. More results can be found in the supplementary material.

4.3. Relighting

Finally, having the full BRDF at hand (albedo, specular, and roughness), we can now do a complete visually accurate rendering of the full scene under new synthetic lighting with additional virtual objects, see Figure 10. To this end, we deploy a path tracing engine with four bounces. The bunny and statue added to the reconstructions of the Office 0 and Office 2 scenes [48] look faithful and realistic as they take the overall scene's appearance into account resulting in consistent shadowing and material effects.

4.4. Limitations and future work

We assume geometry to be given, thus deterioration can have negative impact on the result (Figure 9 Office 1) – a standard limitation for inverse rendering under known geometry [3, 13, 52]. In our tests, we did not experience inter-reflections to cause our system to fail, as target frames are chosen to maximize specular reflections based on direct illumination. However, we expect the presence of strong inter-reflections to limit the performance of our framework, due to the single bounce assumption. In the future, we aim to overcome some limitations with the lit diffuse HDR texture, as it can suffer from remaining baked-in view-dependent effects. While modest corruptions are tolerable and still enable plausible relighting (Section 4.1), we assume the system to not work as assumed when artifacts dominate the median texture.

5. Conclusion

We introduced a method that estimates the BRDF and shading properties of complete 3D scenes from HDR imagery. We are able to recover per surface element albedo and shading using only the reconstructed geometry and HDR textures. We provide a scheme to automatically calculate target frames per object; these are then used to estimate non-diffuse material parameters per object. Numerous experiments on a range of challenging real-world HDR data sets validate the efficiency of our approach compared to the current state-of-the-art, allowing us to create reconstructions that are almost indistinguishable from the real-world.

References

[1] N. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[2] P. Andersson, J. Nilsson, T. Akenine-Möller, M. Oskarsson, K. Åström, and M. D. Fairchild. Flip: a difference evaluator for alternating images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (HPG 2020)*, 3(2), 2020. 7, 8

[3] D. Azinovic, T.-M. Li, A. Kaplanyan, and M. Niessner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 7, 8

[4] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[5] S. Bi, Z. Xu, K. Sunkavalli, D. Kriegman, and R. Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020. 2

[6] M. Brahimi, Y. Quéau, B. Haefner, and D. Cremers. *On the Well-Posedness of Uncalibrated Photometric Stereo Under General Lighting*, chapter Advances in Photometric 3D-Reconstruction, pages 147–176. Springer International Publishing, Cham, 2020. 2

[7] B. Burley and W. D. A. Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7, 2012. 3

[8] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems Conference (RSS)*, June 2013. 3

[9] C. Che, F. Luan, S. Zhao, K. Bala, and I. Gkioulekas. Inverse transport networks. *arXiv preprint arXiv:1809.10820*, 2018. 2

[10] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982. 3

[11] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 189–198, New York, NY, USA, 1998. Association for Computing Machinery. 2

[12] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, 1997. 2, 3, 4

[13] Y. Dong, G. Chen, P. Peers, J. Zhang, and X. Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG)*, 33(6):1–12, 2014. 2, 8

[14] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 2

[15] B. Haefner, S. Peng, A. Verma, Y. Quéau, and D. Cremers. Photometric depth super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2453–2464, 2020. 2

[16] B. Haefner, Z. Ye, M. Gao, T. Wu, Y. Quéau, and D. Cremers. Variational uncalibrated photometric stereo under general lighting. In *International Conference on Computer Vision (ICCV)*, Seoul, South Korea, October 2019. 2

[17] R. Jain and I. Chlamtac. The p^2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Commun. ACM*, 1985. 4

[18] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 2

[19] H. C. Karaimer and M. S. Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[20] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon. Differentiable rendering: A survey. 2020. 2

[21] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Workshop on Rendering Techniques*, pages 103–114. Springer, 2001. 2

[22] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics (TOG)*, 22(2):234–257, 2003. 2

[23] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[24] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 2

[25] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[26] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 2

[27] S. Lombardi and K. Nishino. Radiometric scene decomposition: Scene reflectance, illumination, and geometry from rgb-d images. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 305–313. IEEE, 2016. 2

[28] M. McGuire. Computer graphics archive. https://casual-effects.com/data, July 2017. 8

[29] M. Meilland, C. Barat, and A. Comport. 3d high dynamic range dense visual slam and its application to real-time object re-lighting. In *2013 IEEE International Symposium*

on Mixed and Augmented Reality (ISMAR), pages 143–152. IEEE, 2013. 2

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[31] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 2

[32] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 3

[33] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38(6):1–17, 2019. 2

[34] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich. Optix: A general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4):66:1–66:13, July 2010. 6, 7

[35] G. Patow and X. Pueyo. A survey of inverse rendering problems. In *Computer graphics forum*, volume 22, pages 663–687. Wiley Online Library, 2003. 2

[36] S. Peng, B. Haefner, Y. Quéau, and D. Cremers. Depth super-resolution meets uncalibrated photometric stereo. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017. 2

[37] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 2, 4, 5

[38] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128, 2001. 2

[39] J. Riviere, P. Peers, and A. Ghosh. Mobile surface reflectometry. In *Computer Graphics Forum*, volume 35, pages 191–202. Wiley Online Library, 2016. 3

[40] L. Sang, B. Haefner, and D. Cremers. Inferring super-resolution depth from a moving light-source enhanced rgb-d sensor: A variational approach. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Colorado, USA, March 2020. 2

[41] C. Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994. 3

[42] C. Schmitt, S. Donne, G. Riegler, V. Koltun, and A. Geiger. On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[43] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 2

[44] S. A. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985. 3

[45] J. Shi, Y. Dong, H. Su, and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1694, 2017. 2

[46] B. Smith. Geometrical shadowing of a random rough surface. *IEEE transactions on antennas and propagation*, 15(5):668–671, 1967. 3

[47] The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/. 8

[48] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6, 7, 8

[49] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Josa*, 57(9):1105–1114, 1967. 3

[50] T. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *JOSA*, 65(5):531–536, 1975. 3

[51] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. *Rendering techniques*, 2007:18th, 2007. 3

[52] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 215–224, 1999. 2, 4, 5, 8

[53] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 207–217, 1998. 2

[54] Y. Yu, A. Meka, M. Elgharib, H.-P. Seidel, C. Theobalt, and W. A. Smith. Self-supervised outdoor scene relighting. In *European Conference on Computer Vision*, pages 84–101. Springer, 2020. 2