Eurographics Symposium on Rendering (DL-only Track) (2021)
A. Bousseau and M. McGuire (Editors)

Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering

Merlin Nimier-David[1,2], Zhao Dong[1], Wenzel Jakob[2] and Anton Kaplanyan[1]

[1]Facebook Reality Labs, USA
[2]EPFL, Switzerland

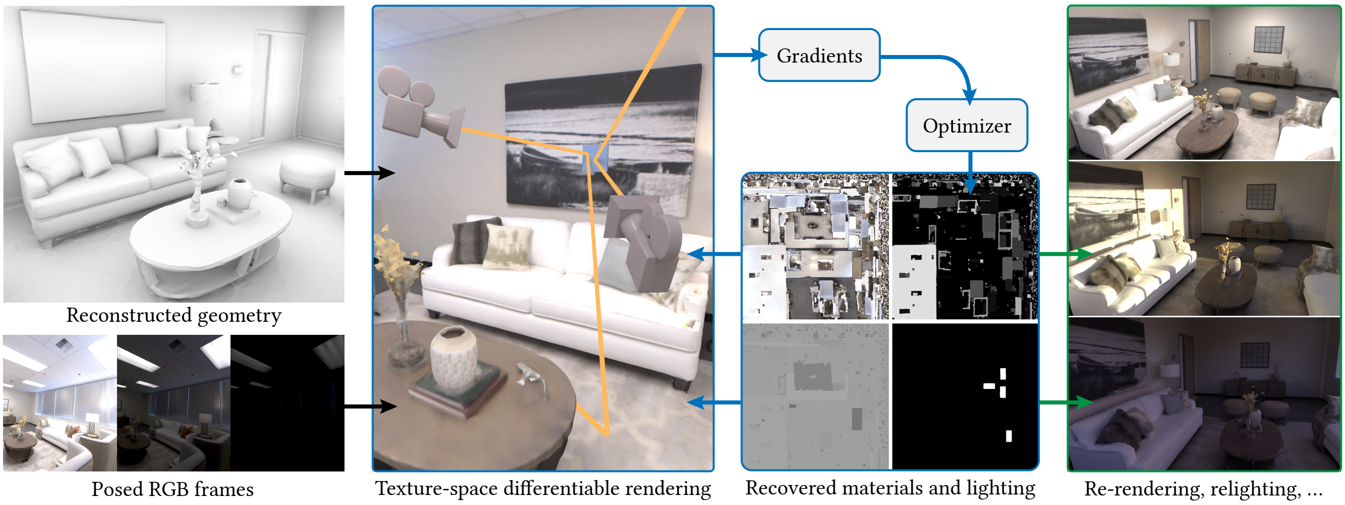

Reconstructed geometry
Posed RGB frames
Texture-space differentiable rendering
Gradients
Optimizer
Recovered materials and lighting
Re-rendering, relighting, ...

Figure 1: *Given posed RGB frames and scene geometry, our method jointly reconstructs lighting and material parameters of real indoor scenes. Our method relies on differentiable rendering, a new texture space sampling scheme as well as carefully designed inductive priors to achieve high quality reconstruction at 4K resolution. The optimized material and lighting parameters are readily used in any physically based graphics pipeline, enabling full scene relighting, re-rendering and AR / VR applications.*

Abstract

Modern geometric reconstruction techniques achieve impressive levels of accuracy in indoor environments. However, such captured data typically keeps lighting and materials entangled. It is then impossible to manipulate the resulting scenes in photorealistic settings, such as augmented / mixed reality and robotics simulation. Moreover, various imperfections in the captured data, such as missing detailed geometry, camera misalignment, uneven coverage of observations, etc., pose challenges for scene recovery. To address these challenges, we present a robust optimization pipeline based on differentiable rendering to recover physically based materials and illumination, leveraging RGB and geometry captures. We introduce a novel texture-space sampling technique and carefully chosen inductive priors to help guide reconstruction, avoiding low-quality or implausible local minima. Our approach enables robust and high-resolution reconstruction of complex materials and illumination in captured indoor scenes. This enables a variety of applications including novel view synthesis, scene editing, local & global relighting, synthetic data augmentation, and other photorealistic manipulations.

CCS Concepts

• ***Computing methodologies*** → ***Reconstruction;*** *Mixed / augmented reality; Virtual reality; Ray tracing;*

1. Introduction

Realistic reconstruction of real 3D environments is a major component of virtual world building. It enables various simulations, augmentations, as well as augmented and virtual reality (AR/VR) applications, such as virtual object insertion, scene relighting, re-rendering from novel views, and material editing. Computer vision techniques have mostly relied on simple lighting, material and light transport models, that do not account for complex illumination, shadows, or view-dependent reflections (Figure 2). With the recent popularity of inexpensive commodity RGB-D sensors and even mobile LiDARs, incredible advances have been achieved for 3D geometry reconstruction [NIH*11, IKH*11, NZIS13, DNZ*17]. For example, a complex room-scale scene geometry with high dynamic range (HDR) textures and semantic labeling can be fully reconstructed in high quality [SWM*19]. However, recovering material or illumination properties requires a deeper understanding of these captured scenes. Limited attention has been devoted to this type of reconstruction, which is a key prerequisite for a seamless photorealistic experience in the aforementioned applications.

Meanwhile, in the computer graphics community, path tracing, a stochastic light transport simulation method has been successfully used to produce photorealistic imagery for movies, visual effects, and games. Path tracing simulates the inter-reflection of light within the scene using a stochastic Monte Carlo integration procedure that accounts for physically based emission and material properties to produce a photorealistic image. Recently, Li *et al.* [LADL18] demonstrated that this entire physically based rendering process can be differentiated, enabling gradient-based optimization of scene properties. Given one or multiple captured RGB frames and an initial guess for scene parameters (including 3D geometry), this approach computes derivatives of an objective function with respect to unknown scene parameters. The objective function can include the difference between the rendered images and the real camera observations. Leveraging these rendering derivatives, the material and lighting parameters are progressively improved by combining the differentiable path tracing process with an optimization technique such as stochastic gradient descent (SGD).

Similar to other non-linear inverse optimizations, strong domain-specific inductive bias and a carefully designed optimization routine are the key to achieving a robust and efficient convergence and high-quality results. For synthetic scenes with perfect geometry and segmentation, the method of Azinović *et al.* [ALKN19] provides high-quality estimates of the scene's materials and lighting. However, only a limited number of proof-of-concept results were shown for real-life captures, with degraded quality. As pointed out by the authors [ALKN19], imperfections in the input data can have significant impact on the quality of the recovered materials.

We propose a method with multiple novel priors to robustly handle large real-world captures and address various imperfections in the input data, such as missing reconstructed geometry, camera misalignment and unevenly distributed camera views. In combination with our new texture-space optimization formulation, our method robustly recovers physically based spatially varying materials and lighting in large captured indoor environments, such as the Replica dataset [SWM*19]. Our key contribution is a robust joint material and lighting reconstruction method that handles:

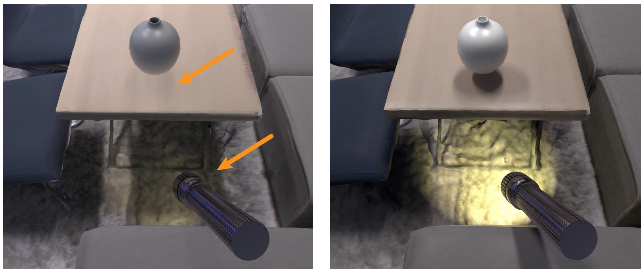


Figure 2: *Simplified solutions for scene reconstruction (left) typically model the world as emissive surfaces without correctly accounting for light transport, and thus do not support important applications such as virtual object insertion and relighting. Our method (right) recovers emission and material parameters of real-world scenes, which are readily used in photorealistic applications.*

- imprecisely reconstructed or even missing geometry
- large number of unevenly distributed live-captured views, with sensor noise, lens distortion, etc.
- estimation of rich materials (multi-lobe BRDF), including high-resolution (4K) textures
- practical run time and memory resources relative to the scale of the optimization problem

The output of our pipeline is a physically based scene representation using graphics industry-standard formats, suitable for photorealistic relighting and re-rendering.

2. Related work

Light estimation. Image-based lighting relies on a recovered HDR environment map [Deb98], which represents the illumination incident from every direction at a single point in the scene, to realistically relight virtual objects at that location. The recovered environment map can be further approximated with certain basis functions, such as spherical harmonics (SH) [RH01, MKC*17] or spherical Gaussians [LSR*20], for efficient rendering. Convolutional Neural Network-based approaches can also automatically estimate an environment map from a single indoor image [GSY*17]. Recent methods [GSH*19, LSR*20] attempt to reconstruct local spatially-varying lighting from a single image. Gardner *et al.* [GHGS*19] utilize a deep encoder network to recover parametric lighting in the scene. In our method, lighting is represented locally as emission from surfaces that are physically present in the scene, e.g. neon lights on the ceiling. The position and radiance of those emitters are determined automatically through our optimization process.

Material recovery. Chen *et al.* [CZS*19] and Kang *et al.* [KCW*18] cast reflectance capture into a form that admits a solution via deep networks. Encoder-decoder architectures are used for appearance capture and rendering of human faces [LSSS18], image-based relighting from sparse samples [XSHR18], and appearance maps [MLTFR19]. Deschaintre *et al.* [DAD*18] use a differentiable re-rendering loss and procedurally generated materials to train a deep network recovering SVBRDF parameters. Gao *et al.* [GLD*19] optimize directly in the latent space learnt by an auto-encoder, which acts as a regularizer. For more details, we refer to the recent survey on deep appearance modeling [Don19]. To the best of our knowledge, none of the existing methods handle non-Lambertian materials, spatially-varying *local* illumination and global light transport all at once.

Material and shape recovery. Schmitt *et al.* [SDR*20] rely on a hand-held RGB-D scanner with active illumination to the reconstruct geometry and SVBRDF of a single object. They use differentiable material clustering to improve estimation of specular components. Several recent works [SC20,BJK*20,LXR*18] use deep cascaded architectures trained on synthetic datasets to recover shape and microfacet SVBRDF of a single object from one or two handheld pictures. Li *et al.* [LXR*18] additionally account for global illumination with dedicated neural blocks. In contrast, our method scales to large indoor scenes with significant interreflection.

Joint estimation of material and lighting. Barron *et al.* [BM14] recover geometry, reflectance, and illumination from a single image of an arbitrary object by enforcing hand-crafted priors on each component. Li *et al.* [LSR*20] similarly recover depth, SVBRDF and local illumination from a single viewpoint using a deep network trained on realistic synthetic interior scenes. Karsch *et al.* [KHFH11, KSH*14] render synthetic object into real photos. Zhang *et al.* [ZCC16a] achieve plausible results at recovering the reflectance of walls, floor, and ceiling of indoor scenes along with lighting using inverse rendering.

Azinović *et al.* [ALKN19] is a step towards the general use of differentiable rendering for reconstruction, though it remains far from achieving this goal for real captures with imperfect input data. Our method builds on this approach, while supporting complex spatially-varying materials that are reconstructed from real captured data.

Intrinsic decomposition. Image-space methods [BTHR78, BM14, DAD*18, MMZ*18, LGZ*20] employ sophisticated data-driven approaches, by learning the distributions of material and illumination. However, these methods do not have a notion of 3D geometry, and cannot handle occlusion, interreflection, and physically based factors such as the squared distance falloff of light intensity. They also require a significant amount of training data, and are prone to errors outside of the training data set.

Deferred neural rendering. Deferred neural rendering [TZN19] achieves novel view synthesis, scene editing, animation synthesis, or free viewpoint relighting [GCD*20] by optimizing a neural texture jointly with a neural renderer. The high-dimensional neural texture, mapped to a simple 3D proxy surface, is sampled as in the standard graphics pipeline to produce features that are decoded into an image by the neural renderer. In contrast, our method produces standard physically based textures that are readily used in traditional renderers.

Stratified sampling. Stratified sampling [Coo86] is a well-known techniques used in Monte Carlo rendering to reduce variance over purely uniform sampling. In the context of non-line-of-sight geometry reconstruction, Tsai *et al.* [TSG19] sample over the surface of the recovered shape, in order to improve rendering efficiency (fewer missed rays) as well as to produce more coherent ray bundles. In the same spirit, our texture space sampling technique improves convergence by sampling uniformly in the space of optimization variables rather than generating rays in camera space.

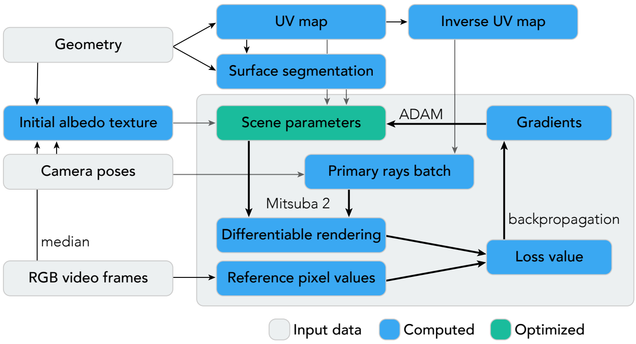


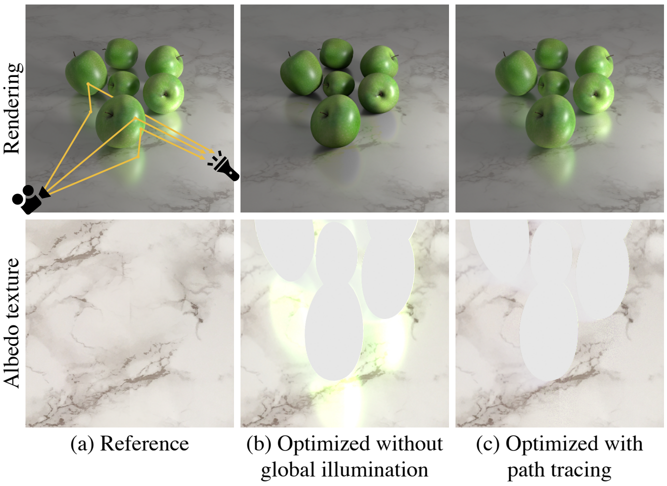
Figure 3: *Our reconstruction technique based on differentiable rendering recovers complex spatially varying materials and light sources from posed handheld RGB frames. A novel texture-space sampling scheme robustly handles uneven coverage and imperfections in real-world data.*

Differentiable rendering. Blanz and Vetter used differentiable rendering for face reconstruction [BV99]. Inverse radiosity (*e.g.*, [YDMH99,ZCC16b]) achieves impressive results for solving near-field illumination and Lambertian materials for indoor scenes. Gkioulekas *et al.* [GZB*13, GLZ16a] and Che *et al.* [CLZ*18] solve for scattering parameters using a differentiable volumetric path tracer. Kasper *et al.* [KKSH17] developed a differentiable path tracer, but focused on distant illumination. Loper and Black [LB14] and Kato [KUH18] developed fast differentiable rasterizers, but do not support global illumination. Physically based differentiable rendering [LADL18, NDVZJ19, LHJ19] made it possible to compute derivatives of an entire physically based light transport simulation, including global illumination, with respect to the unknown parameters of the rendered scene. We use the Mitsuba 2 [NDVZJ19] differentiable path tracer in our optimization pipeline.

3. Method

Our method uses an analysis-by-synthesis approach: at each step, images of the scene with current parameters are obtained using differentiable rendering and compared to the observed reference. The difference is then minimized using gradient-based optimization. Figure 3 shows the high-level pipeline discussed in the remainder of this section.

Unlike prior work based on rasterization, the rendering step of our method builds on a differentiable path tracer and physically based appearance and illumination models. The resulting images account for global illumination, which is a prerequisite for high-fidelity parameter reconstruction, as illustrated in Figure 4. Many



(a) Reference (b) Optimized without global illumination (c) Optimized with path tracing

Figure 4: *Real-world light transport features global illumination (GI) between objects. Reconstructing the marble table texture in this synthetic scene* ***(a)*** *poses severe challenges for classic techniques. The reflection of the apples cannot be explained without GI* ***(b)****, and this discrepancy between reality and reconstruction can only be explained via incorrect color bleeding. A differentiable path tracer* ***(c)*** *can disentangle the effects caused by individual objects. In both cases, unobserved texture regions remained at their initialization value (gray).*

open-ended steps are necessary to convert this high-level idea into a practical and robust algorithm for inverse rendering, and the main contribution of our approach lies in those specific design decisions that we now explain in more detail.

3.1. Input data

The input to our method is a reconstruction of the scene geometry and a set of RGB photographs with camera intrinsics and extrinsics ("posed frames"). We further use an approximate segmentation of the scene's surfaces, which can either be computed automatically or provided as input (see supplemental document). Our method reconstructs suitable material and lighting parameters, but does not modify the input geometry — joint material and geometry reconstruction is in principle possible thanks to recent advances in differentiable rendering [LADL18], but is out of scope of this work.

We use the recent *Replica* dataset [SWM*19] in our examples, which consists of multiple indoor scenes acquired using a handheld device. The resulting data was processed using standard methods producing camera extrinsics and intrinsics, a triangular mesh of the scene, and an approximate instance segmentation. Our method is not specific to the Replica dataset. That said, physically-based material and lighting reconstruction requires sufficient *dynamic range* of observations, e.g., to provide some direct observations of light sources and highlights without overexposure. The Replica dataset provides this via alternating exposures of the RGB frames (*multiplexed HDR*).

Regardless of the source, real-world data invariably contains noise and imperfections such as imprecise camera poses, surface normals, missing fine geometry, and inexact segmentation, producing systematic discrepancies between renderings and the observations. It is imperative that the method handles such flaws gracefully.

In our pre-processing step, we discard frames with severe under- and over-exposure and motion blur, which are easily detected from the difference in camera pose between adjacent frames. We also linearize and white-balance the images, and remove lens distortion. Scene geometry is represented with a standard triangle mesh with UV texture parameterization. The original Replica data lacks UV coordinates, so we generate them automatically using Blender's standard "Smart UV Project" operator [Com19].

3.2. Inductive bias & physically based assumptions

The inverse problem targeted by our method is highly ambiguous: each surface location within the scene can in principle affect the color of any other position via indirect reflections. Because light emitted by a light source can interact with multiple materials before arriving at the camera, any given observation can be explained in multiple ways. For example, objects seen via specular reflection can be misattributed as emission or diffuse reflectance (Figure 4).

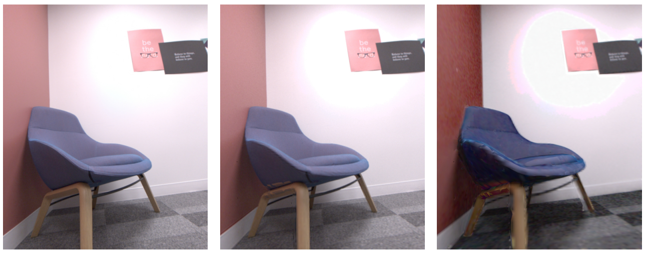
Therefore, a naive application of image-based differentiable rendering systematically overfits with poor local minima. We introduce several inductive biases that promote plausible and consistent results to address these issues.

Emitters. We model light sources as area lights (emissive surfaces) with a cosine-based directional profile, which are standard in physically based rendering. However, emission is difficult to disentangle from reflection. For example, a highlight observed on a surface can be misinterpreted as a light source. In our experiments, unrestricted optimization always converges to implausible solutions with spatially varying emission on all scene surfaces (Figure 5).

Therefore, we initially restrict spatial variation of emission to a single intensity value per object, based on the instance segmentation. Once light sources have been identified by the optimization, we enable spatially-varying emission over those regions.

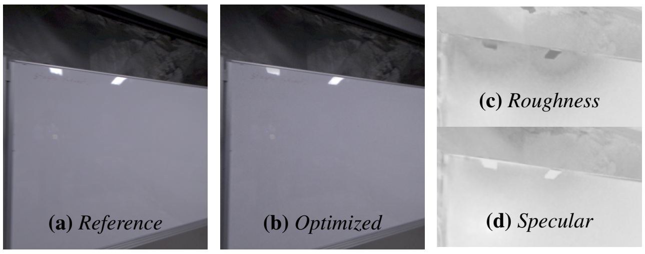
Material model. In physically based rendering, the bidirectional reflectance distribution function (BRDF) models the surface material and defines how much radiance is reflected from an incident direction to an outgoing direction. A large variety of general and specialized BRDF models have been proposed throughout decades of research [GGG*16].

Since we do not assume access to a classification of materials over the scene, we choose a single material model that can cover the majority of appearance, while keeping the number of parameters to a minimum. Spatial variations are handled by texturing the model's parameters. The Disney BRDF [BS12] is a widely adopted material model used in movies and games that captures a versatile set of appearance with intuitive controls. It exposes ten high-level parameters such as base RGB color, metalicness, roughness, and clearcoat. In unstructured optimization, however, different parameter configurations often lead to similar appearance. To reduce this ambiguity, we restrict optimization to only handle opaque surfaces with diffuse albedo, roughness, and specular parameters.



(a) *Reference* **(b)** *Optimized (re-rendering)* **(c)** *Optimized (emission only)*

Figure 5: *Unconstrained joint optimization of material and emission reproduces the reference **(a)** with high fidelity, as shown in **(b)**. However, the solution found by the optimizer is absurd, since it turns the entire scene into an emissive surface **(c)**. Since no light sources are visible in that frame, a correct reconstruction would have **(c)** be entirely black.*



(a) *Reference* **(b)** *Optimized* **(c)** *Roughness* **(d)** *Specular*

Figure 6: *Unconstrained optimization of material parameters leads to implausible results, such as materials being highly glossy only where highlights were observed.*

A second important ambiguity is due to the fact that we can judge about a surface's roughness at a point only when we observe an actual highlight at that point. Therefore, the optimizer is able to infer surface specularity and roughness *only where* the specular highlights are observed (Figure 6). However, we cannot expect our captured data to observe highlights at every point they could occur. Therefore, we assume the roughness and specularity to remain constant within a class of the surface segmentation and optimize a single roughness & specularity value per object. This is a common case in real world and also reduces the number of parameters to optimize, speeding up convergence.

Initialization. Proper initialization is important to guide optimization toward a good minimum. We start with neutral constant values for the emission (0), roughness (0.5) and specular (0.5) parameters. To initialize the spatially-varying diffuse color, we build a median texture by iterating over all available reference pictures and projecting them in texture space. We employ an online median estimator [FS07] to avoid memory concerns when given tens of thousands of views. We preferred the median to the mean filter to suppress view-dependent effects such as specular reflection.

3.3. Texture-space sampling for variance reduction

A direct application of differentiable rendering is to render the current reconstructed scene from a given view, and minimize the distance to all pixels in the observed image at the same view. However, the target unknowns (material and illumination parameters) lie in the space of scene surfaces (texture space). Therefore, optimizing over all pixels within a frame(s) lead to uneven convergence, since the unknowns (e.g. albedo texture values) are observed unevenly within a single view, as well as over the video sequence (visualized in Figure 7). Additionally, view-dependent effects such as glossy highlights require multiple observation angles to disambiguate the roles of diffuse and specular components.

Texture-space sampling Instead, we propose to form the training batches by sampling the unknowns uniformly *directly* in texture space. This is more efficient than sampling a random subset of views, as it allows to reduce the noise in gradients by proceeding with batches of observations that are directly relevant to the selected unknowns.

Texture-space sampling is realized as follows (pseudocode is given in the supplemental document). At the start of each iteration, we select a subset of the unknown variables by sampling uniformly at random over texture space. The number of sampled points is effectively a batch size, which can be adjusted based on the available GPU memory. Using a precomputed inverse UV mapping, we lookup the corresponding 3D positions on the scene surfaces. Next, we connect the sampled mesh positions to a set of reference view positions. Connections that are occluded by geometry, or simply fall outside of the cameras' frusta, are discarded. For this batch of visible 3D positions, we fetch the corresponding pixel values from the reference RGB frames. Finally, we estimate the current radiance values for the batch with differentiable path tracing. After computing the per-pixel loss (averaged over all rays), gradients are obtained by backpropagating through the rendering algorithm. Finally, scene parameters are updated with an optimizer step.

Jacobian factors Transformations applied to the samples–mapping from UV space to scene surfaces and finally to camera rays–imply a change of probability density. In standard Monte Carlo rendering, that change should be accounted for when computing the sampling weight by multiplying it with the Jacobian determinant of each transformation [VG95, PJH16]. These factors include a term accounting for the UV mapping's distortion, as well as a geometry term $\frac{\cos(\theta)}{d^2}$, with θ the incident angle to the sampled surface and d the distance to the camera. Intuitively, the geometry term corrects for the fact that sampling that same surface point from the camera's directional distribution becomes less likely as the distance increases, or the observation angle more grazing.

However, the explicit goal of our sampling technique is to assign equal weight to all optimization variables. Note that in the context of an optimization, we are free to define the objective function as needed to improve convergence and reconstruction quality. To this end, we omit the Jacobian terms above, implicitly introducing a factor α_j cancelling them out in our per-ray objective function (Equation 1).

This is in contrast with the method of Tsai *et al.* [TSG19], where reconstructed surface points are sampled directly and all Jacobian terms are included. In their non-line-of-sight reconstruction application, observation distances are roughly constant, while ours vary

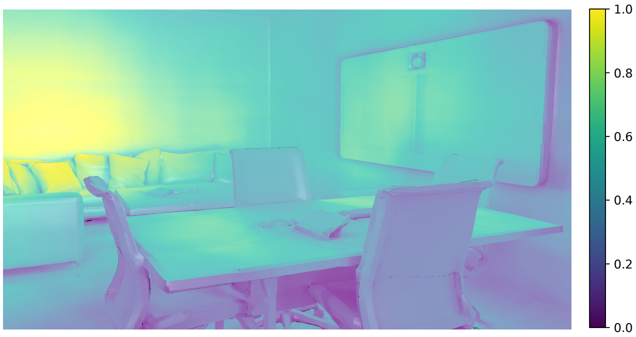
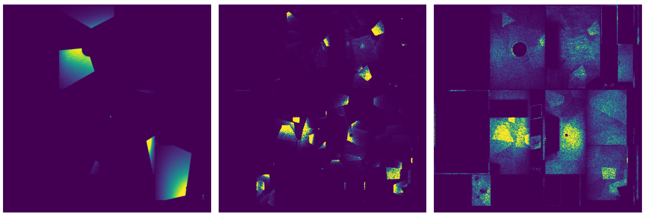


Figure 7: *Unstructured capture of real-life scenes from handheld video results in uneven density of observations, which in turn leads to uneven convergence with naive inverse rendering. We visualize the relative number of observations for each location of the* OFFICE-2 *scene [SWM* 19].*



(a) *Image-space* **(b)** *Texture-space* **(c)** *Texture-space (re-weighted)*

Figure 8: *We illustrate the gradients produced by different sampling techniques. We visualize albedo gradient magnitude in texture space directly at a given iteration. Standard image-based differentiable rendering **(a)** leads to nonzero gradients only within the frusta of cameras selected for this iteration. Our texture-space sampling scheme **(b)** selects texels uniformly at random and connects them to camera positions, resulting in even coverage within an iteration. Finally, we implicitly reweight the objective function by omitting Jacobian factors **(c)** in order to obtain gradients of comparable magnitude regardless of observation distance or angle. Note that some locations are not mapped to any scene surface and thus have zero gradients.*

greatly from viewpoint to viewpoint. Omitting Jacobian factors also helps us avoid exploding gradients when d approaches zero. Finally, we obtain gradients of comparable magnitude spread evenly over texture space, as illustrated in Figure 8.

3.4. Optimization details

Coarse-to-fine optimization. Due to the ambiguities described in Section 3.2, starting the optimization with all unknowns of a large scene at their highest resolution leads to low-quality local optima. “Coarse-to-fine” schemes have been shown to help greatly in similar cases [GLZ16b, NDVZJ19, NDSRJ20].

In that spirit, we gradually introduce optimization variables: first emission, then roughness & specular coefficients, and finally spatially-varying albedo. In spirit of texture MIP pyramids, the albedo is first optimized as a 1024 × 1024 RGB texture, and then refined in two stages to reach the final 4096 × 4096 resolution. After the first stage, emitters contributing a small fraction of the total scene radiance are rounded to zero.

Any subset of steps can be repeated as needed by restarting the optimization from the previous step's results. In our experiments we found this to be unnecessary, since a single run through this sequence yields good convergence.

Loss function. We use a pixel-wise mean squared loss. The loss operates in linear color space, i.e., without gamma compression. Under-exposed (resp. over-exposed) values are handled with a one-sided difference that only penalizes values above (resp. below) the clipping threshold $v_{\min}$ (resp. $v_{\max}$). The optimization formulation is therefore:

$$\min_{\theta} \; \frac{1}{n} \sum_{j=1}^{n} \alpha_j w_j \left(\hat{I}_j - I_j\right)^2, \tag{1}$$

$$w_j = \mathbb{1}_{\hat{I}_j \geq v_{\min} \cup I_j \geq v_{\min}} \cdot \mathbb{1}_{\hat{I}_j \leq v_{\max} \cup I_j \leq v_{\max}},$$

where θ comprises all unknown scene parameters, $\hat{I}_j$ is the estimated radiance along ray j, I_j is the reference observed pixel value corresponding to the ray, $\mathbb{1}$ is the indicator function, w_j encodes the one-sided difference, and α_j is the implicit reweighting term described in Section 3.3.

Optimizer. We use the Adam optimizer [KB14] with learning rates 1 for emission, 0.005 for roughness & specular, and 0.1 for diffuse albedo. Other parameters are set to their recommended defaults.

We found two additional modifications to be important. First, recall that in addition to the noisy stochastic gradient descent, each path-traced sample is itself a noisy Monte Carlo estimate. The path-traced sample values often include extremely high-valued outliers caused by improbable light paths. Therefore, we regularize this signal by clamping gradients to $\pm 10^{-8}$ (before Adam rescaling) to prevent these outliers from contaminating the optimized textures.

Another important observation is that at each iteration, only a subset of the scene is observed and can receive meaningful gradients. Moreover, the moment estimates maintained by Adam inevitably include Monte Carlo stochastic estimation noise from previous iterations' gradients. As a result, at each step of the optimizer, noisy momentum is applied repeatedly to all variables, even those not observed in the current batch. As long as no new observations are made for a given variable, the noise pattern in its momentum remains fixed, impeding convergence. To alleviate this issue, we restrict the application of momentum within the optimizer, as well as updates to the moments, to variables which receive nonzero gradients at that iteration.

Discarding indirect gradients. Using differentiable path tracing to compute global illumination, we found gradients for indirectly-observed parameters to be extremely noisy due to the low sampling probability of long light paths. Understandably, an observation of a wall tells us relatively little about the opposite wall, even though some indirect light has likely come from there. When the available data allows, it is preferable to rely on direct observations, in order to minimize variance. Therefore, we exclude indirect light bounces from the gradient computation. This results in memory and computational savings as well as faster convergence due to the reduced noise in gradients. Note that we still compute the correct path traced solution, so global illumination is fully accounted for: the net effect is simply to restrict gradient-based updates to regions that are directly observed in a given iteration, while disentangling indirect effects.

Averaging of iterates. Optimization eventually wanders around the true solution due to Monte Carlo noise in the rendered images. We apply Polyak-Ruppert averaging [PJ92, Rup88] by maintaining a running average of the parameter values over the last 10% of the optimization.

4. Results

We now evaluate our method on challenging real-world scenes and against previous work. Additional results, including animated sequences, validation on synthetic data, a comparison to the method of Li *et al.* [LSR*20], and a study of sensitivity to the input data quality are available in the supplementary material.

4.1. Reconstruction of real captured scenes

We apply our reconstruction pipeline to the *Replica* dataset [SWM*19], which includes reconstructed geometry, an approximate instance segmentation, and posed reference images captured with a handheld rig. While we have used the provided instance segmentation for convenience, we show in the supplemental document that a naively-generated segmentation performs comparably well (see supplemental Section 4). This input has imperfections, including imperfect camera registration and missing detailed geometry, which make robust reconstruction challenging.

In order to compare against captured ground truth, we re-render the scene from known viewpoints after our reconstruction is finished. Our re-rendered images match the captured frames closely, including fine textured details and view-dependent effects, as shown in Figure 9. The reconstructed scenes do not overfit to the training views, as shown in the out-of-distribution re-renderings, animated results and comparisons given in the supplemental video.

4.2. Comparison to prior work

We compare to the state-of-the-art method of Azinović *et al.* [ALKN19] for joint materials and lighting estimation in Figure 10. We modify the authors' implementation to support Replica's multiplexed HDR captured images. Spatially-varying parameters are supported in their method by subdividing the geometry and assigning one material per triangle. The optimization is run with the settings recommended in the paper for 6 million iterations, on the same input data as ours.

Their method only accounts for the first two bounces of light transport and must thus produce an overly bright base color to fit the reference, which is especially apparent in shadowed regions, where most light comes from indirect reflection. Roughness and specular, as well as other BRDF parameters (not shown) are optimized freely, which leads to implausible high-frequency variations across surfaces. Finally, a significant amount of Monte Carlo noise is present in the optimized spatially-varying parameters.

In contrast, our robust pipeline allows us to simulate full global illumination with a large number of light bounces (we use 8 in practice as the contribution from further bounces is minimal). Combined with our texture-space sampling method and inductive biases, our method produces plausible and noise-free results with more precisely reconstructed albedo, material parameters, and illumination. Finally, shadows are effectively removed from the albedo texture, and view-dependent effects such as specular highlights are correctly attributed to material parameters.

4.3. Ablation study

In order to evaluate the impact of each of our method's component and design decision, we have conducted a detailed ablation study. We reconstruct emission and material parameters of the OFFICE-0 scene using 9 variants of our method, progressively adding the features described in Section 3. We run each variant for 130 minutes and compute the pixel-wise Mean Squared Relative Error (MSRE) with respect to a fixed set 30 reference images chosen at random (Figure 11, left). Each feature improves the re-rendering loss and / or helps achieve more plausible results. For visual inspection, a crop of the optimized diffuse albedo texture of each variant is shown in Figure 11 (right).

The baseline **(a)** uses image-based optimization, the most direct and ad hoc application of differentiable rendering. Variant **(b)** uses our novel texture-space sampling method, described in Section 3.3. Variants **(c)** and **(d)** add our parametrizations of emitters and materials respectively (Section 3.2). The inductive bias on materials does not decrease error in this experiment, but does ensure more plausible results. Understandably, unconstrained optimization may achieve good error by overfitting, but produce implausible material parameters (see Figures 5 and 6). Variant **(e)** initializes the diffuse albedo parameter with the median of all observations. Variant **(f)** clamps gradients to prevent Monte Carlo noise from contaminating the textured parameters (Section 3.4), while variant **(g)** additionally prevents the Adam state updates and momentum to be applied to variables that were not observed in the current iteration. Variant **(h)** applies Polyak-Ruppert averaging [PJ92, Rup88]. Finally, variant **(i)** uses a coarse-to-fine scheme, progressively introducing degrees of freedom to the optimization (Section 3.4).

4.4. Implementation

Our implementation is based on the Mitsuba 2 differentiable renderer [NDVZJ19], but our method could be implemented in any framework providing the relevant derivatives. Each optimization runs for 12 hours on average on a single NVIDIA Titan RTX GPU with our research implementation. Orders of magnitude improvements in training speed are possible with an optimized implementation [NDSRJ20].

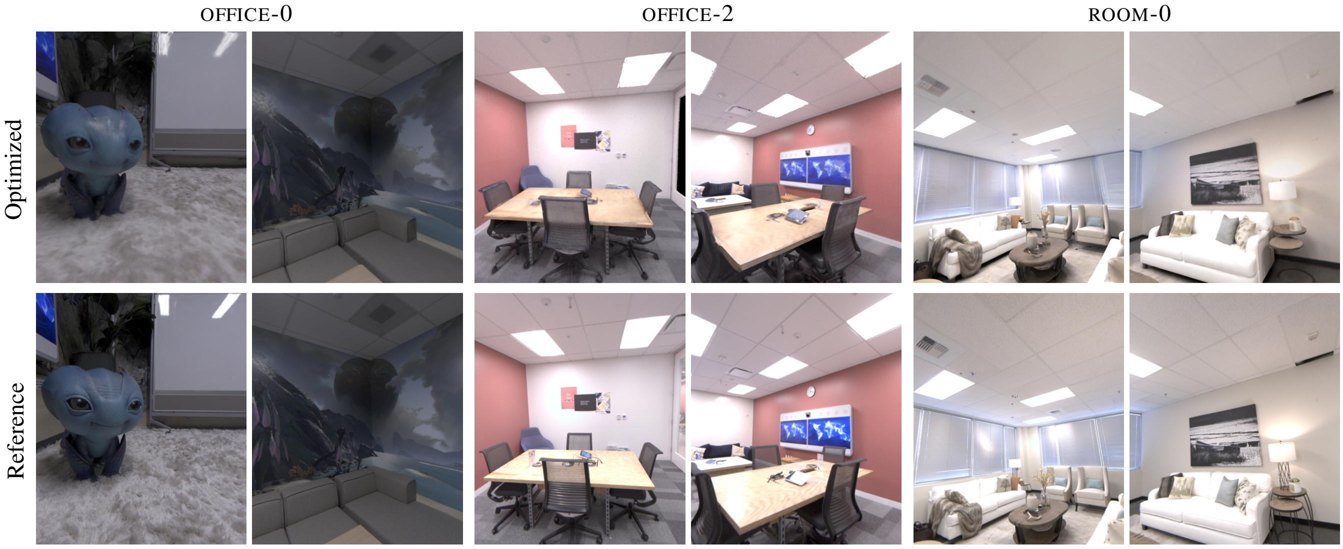


Figure 9: *Re-rendering scenes from the Replica dataset [SWM*19] using the materials and emission parameters recovered by our method matches the reference images closely, including view-dependent effects and high-frequency detail.*

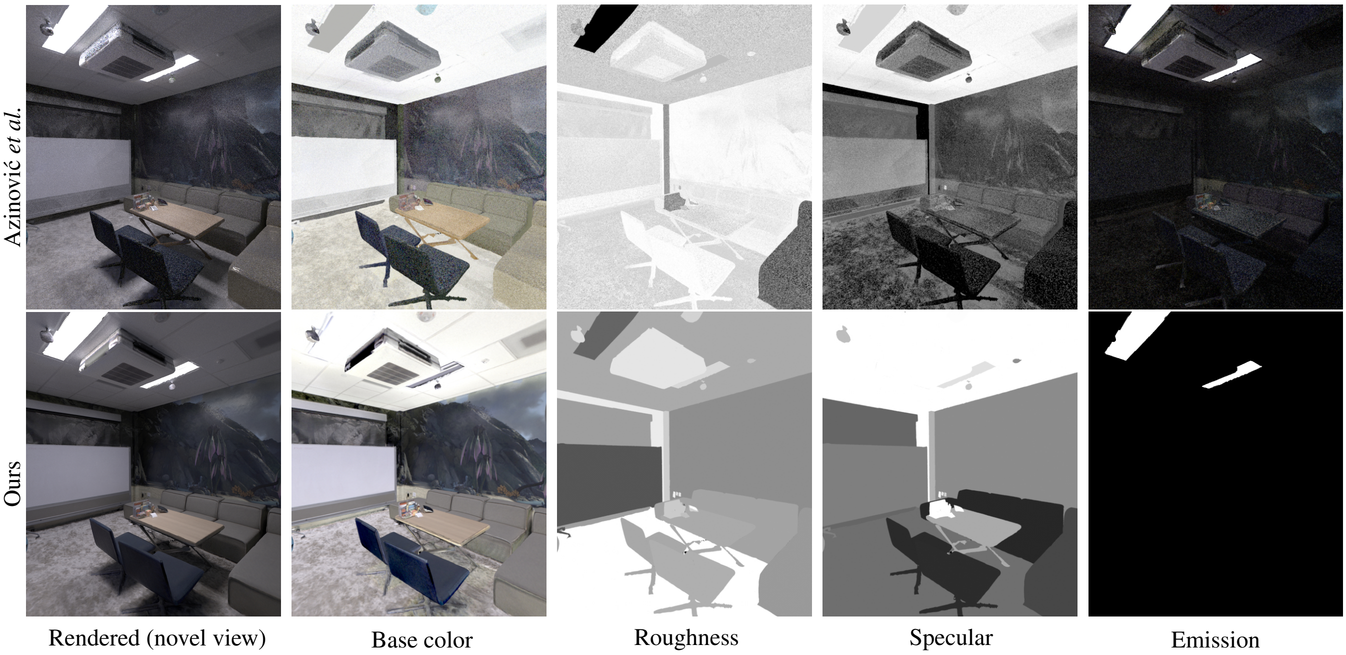


Figure 10: *Given the same inputs, previous work based on differentiable path tracing [ALKN19] outputs textures contaminated by Monte Carlo rendering noise and exhibits several of the issues outlined in Section 3, including uneven convergence and implausible high-frequency changes in roughness & specular material parameters.*

4.5. Applications

Scenes reconstructed with our method generalize well to out-of-distribution views and can be rendered from any previously unobserved viewpoint (Figure 13). We additionally visualize our method's outputs: a set of textures representing the scene's emission and physically based material parameters (diffuse albedo, roughness, and specular). They are well disentangled and noise-free despite significant Monte Carlo noise present during optimization and dataset imperfections. In this format, the scene is ready for use in standard rendering pipelines for photorealistic applications such as scene editing, novel view synthesis, and relighting. Figure 12 (left) demonstrates embedding four virtual objects in the scene, which is a common task for mixed reality applications. Without any additional processing or manual work, the inserted objects

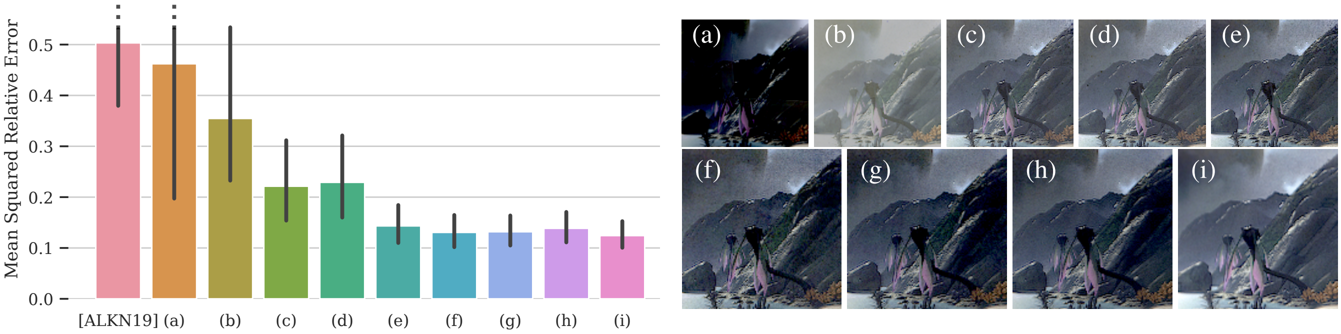


Figure 11: *Ablation study: we progressively add features to a naive inverse rendering baseline* ***(a)*** *up to our full method* ***(i)****. The method of Azinović* et al. *[ALKN19] is included for comparison. The rightmost features result in mostly qualitative improvements, that are visualized with crops of the optimized albedo texture. Our full method* **(i)** *recovers the most details while avoiding noise in the texture entirely.*

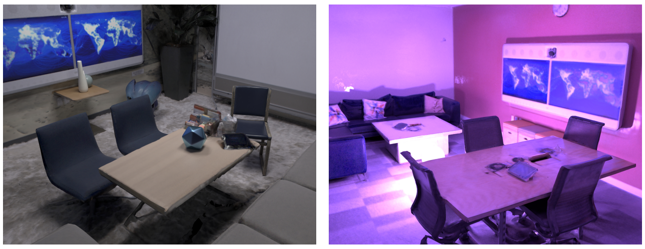


Figure 12: *Using our recovered emission and materials, adding virtual objects to the* OFFICE-0 *scene (left) automatically results in correct shadows, reflections and indirect illumination. Scenes can additionally be relit with arbitrary light sources and rendered from any viewpoint (right).*

blend in and interact correctly with their surrounding (e.g., see reflections, shadows, matched lighting).

Figure 12 (right) shows a complete re-lighting of the scene: existing illumination was removed and a brightly colored light source was added near the floor. The scene reacts correctly to the new illumination, and there are no visible residuals of the original illumination the scene was captured in (such as baked shadows or highlights). Note that this would not be possible without correctly disentangling materials and lighting, as shown in Figure 2.

We believe these and many other possible applications enable more seamless integration of real and virtual worlds in scenarios like virtual and augmented reality, robotics simulation, and dataset augmentation. Please refer to the supplemental video for an animated version of these results.

4.6. Limitations

The main limitation of our method is its requirement of the input reconstructed geometry. In our current scene parametrization, rendered images may only explain observations where geometry is present. For example, if a highly emissive or specular object is entirely missing from the geometry, its contribution will most likely be outprojected and attributed to the background objects by the optimizer. This behavior can be seen in ROOM-0 (Figure 13). Automatically detecting and adding missing emitters would be a valuable improvement in future work.

Our method does not currently handle reconstruction of transparent objects, even if they have correct reconstructed geometry (which is a challenge in itself). An example is shown in Figure 14: the back of the chair is incorrectly assigned the color of the table that should have been seen through it. Differentiable rendering is generally well suited to support advanced effects such as transparency and refractions, as the corresponding light transport is well understood. However, we have limited our light transport simulation to the most common and important effects in order to reduce the underlying optimization complexity and improve robustness of the method in common scenarios.

Finally, current automatic differentiation-based differentiable renderers, such as Mitsuba 2, can consume significant amounts of GPU memory when accounting for global illumination. Ongoing research on efficient and scalable differentiable rendering [NDSRJ20, VSJ21] directly benefits our method by removing this bottleneck.

5. Conclusion and future work

We presented a robust method for material and lighting reconstruction in large captured environments based on differentiable rendering. In order to gracefully handle the unavoidable capture & inputs imperfections, uneven coverage of the reference images, as well as correctly disentangling spatially-varying material and illumination parameters, we introduced a novel texture-space optimization scheme and carefully chosen inductive biases, which guide the reconstruction toward high-quality minima. We believe our method provides an important stepping stone towards full scene understanding, which opens up new opportunities for scenarios where realism is important, such as augmented and mixed reality, robotics and sensory simulation, and synthetic augmentation of datasets.

Differentiable rendering is a powerful paradigm, allowing our method to be naturally extended in future work to reconstruct a wider range of appearance (e.g., transparent and refractive surfaces), as well as illumination from outdoor scenes (e.g., using an environment map). Finally, the imperfections of geometric reconstruction and reference images (such as camera pose, motion blur,

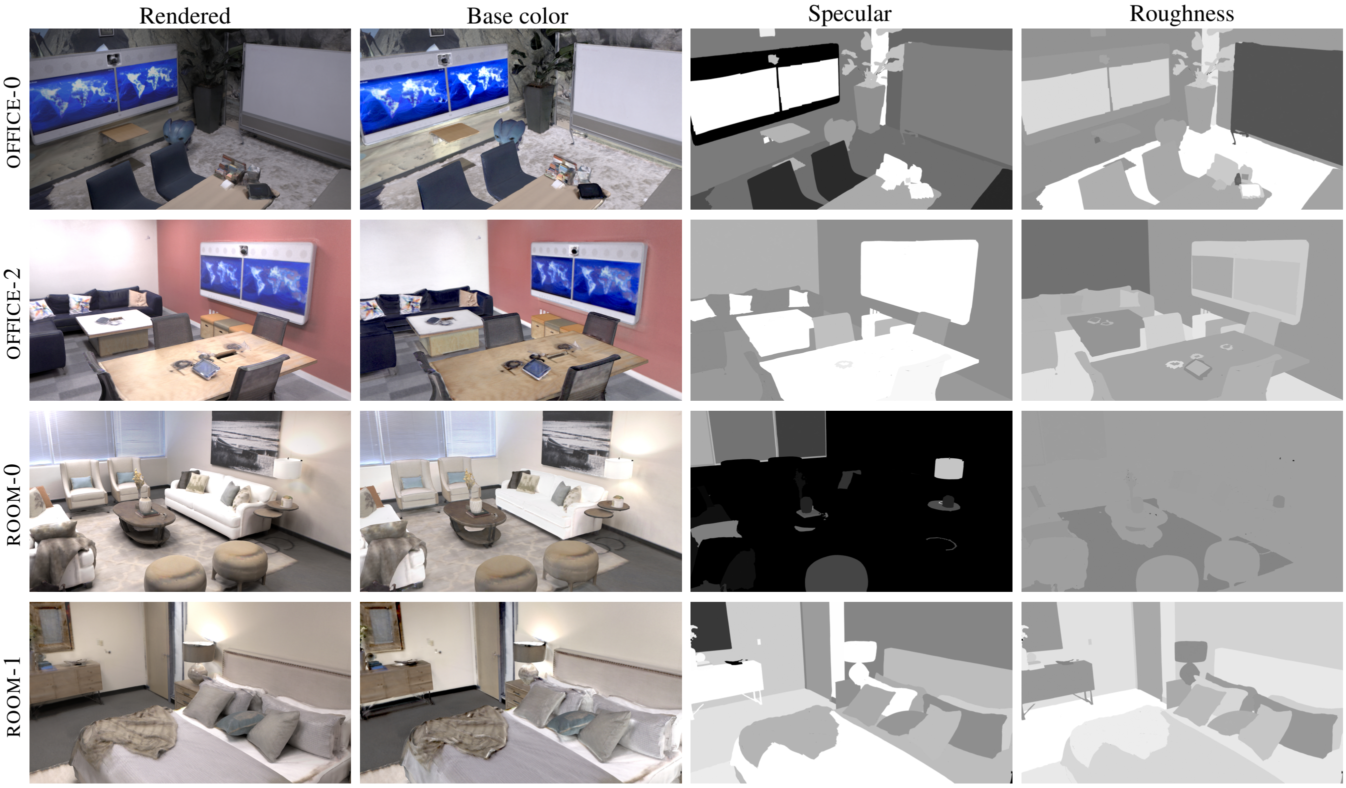
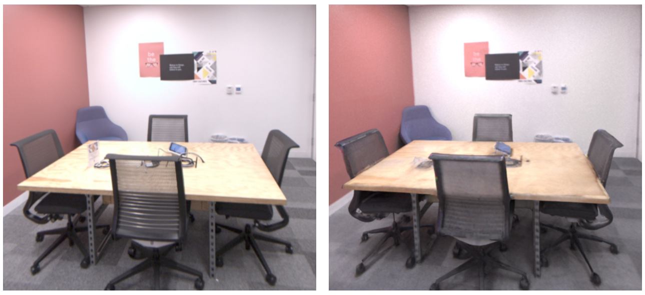


Figure 13: *Scenes obtained with our method can be re-rendered from any viewpoint. The base color texture includes fine detail* (4096 × 4096 *resolution) and all recovered parameters are physically based and correctly disentangled.*



(a) *Reference* **(b)** *Optimized*

Figure 14: *Since the reflectance model used by our method does not support transparency, the color of the table seen through the back of the chair is incorrectly attributed to the chair itself.*

sensor noise, etc) could be integrated to the differentiable simulation and minimized to further improve reconstruction.

Acknowledgements. We thank the anonymous reviewers for their helpful feedback. We are grateful to Dejan Azinović and Tzu-Mao Li for helping run comparisons to their work [ALKN19], as well as Matthew Chapman for creating scene editing examples. The apples model (Figure 4) was released by Alex Telford and the flashlight model (Figure 2) by BlendSwap user *richanatario*.

References

[ALKN19] Azinović D., Li T.-M., Kaplanyan A., Niessner M.: Inverse path tracing for joint material and lighting estimation. In *Proc. of CVPR* (2019), IEEE, pp. 2447–2456. 2, 3, 7, 8, 9, 10

[BJK*20] Boss M., Jampani V., Kim K., Lensch H., Kautz J.: Two-shot spatially-varying brdf and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 3982–3991. 3

[BM14] Barron J. T., Malik J.: Shape, illumination, and reflectance from shading. *IEEE Trans. Pattern Anal. Mach. Intell. 37*, 8 (2014), 1670–1687. 3

[BS12] Burley B., Studios W. D. A.: Physically-based shading at disney. In *ACM SIGGRAPH* (2012), vol. 2012, pp. 1–7. 4

[BTHR78] Barrow H., Tenenbaum J., Hanson A., Riseman E.: Recovering intrinsic scene characteristics. *Comput. Vis. Syst 2* (1978), 3–26. 3

[BV99] Blanz V., Vetter T.: A morphable model for the synthesis of 3d faces. In *SIGGRAPH* (1999), pp. 187–194. 3

[CLZ*18] Che C., Luan F., Zhao S., Bala K., Gkioulekas I.: Inverse transport networks. *arXiv preprint arXiv:1809.10820* (2018). 3

[Com19] Community B. O.: *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2019. URL: `https://www.blender.org`. 4

[Coo86] Cook R. L.: Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG) 5*, 1 (1986), 51–72. 3

[CZS*19] Chen L., Zheng Y., Shi B., Subpa-asa A., Sato I.: A microfacet-based model for photometric stereo with general isotropic reflectance. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019). 2

[DAD*18] Deschaintre V., Aittala M., Durand F., Drettakis G., Bousseau A.: Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph. 37*, 4 (2018), 128:1–128:15. 2, 3

[Deb98] Debevec P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH* (1998), pp. 189–198. 2

[DNZ*17] Dai A., Niessner M., Zollhöfer M., Izadi S., Theobalt C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph. 36*, 4 (2017), 1. 2

[Don19] Dong Y.: Deep appearance modeling: A survey. *Visual Informatics* (2019). 3

[FS07] Feldman D., Shavitt Y.: An optimal median calculation algorithm for estimating internet link delays from active measurements. In *2007 Workshop on End-to-End Monitoring Techniques and Services* (2007), IEEE, pp. 1–7. 5

[GCD*20] Gao D., Chen G., Dong Y., Peers P., Xu K., Tong X.: Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–15. 3

[GGG*16] Guarnera D., Guarnera G. C., Ghosh A., Denk C., Glencross M.: Brdf representation and acquisition. *Computer Graphics Forum 35*, 2 (2016), 625–650. 4

[GHGS*19] Gardner M.-A., Hold-Geoffroy Y., Sunkavalli K., Gagné C., Lalonde J.-F.: Deep parametric indoor lighting estimation. In *Proc. of ICCV* (2019), IEEE, pp. 7175–7183. 2

[GLD*19] Gao D., Li X., Dong Y., Peers P., Xu K., Tong X.: Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–15. 3

[GLZ16a] Gkioulekas I., Levin A., Zickler T.: An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *ECCV* (2016), pp. 685–701. 3

[GLZ16b] Gkioulekas I., Levin A., Zickler T.: An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision* (2016), Springer, pp. 685–701. 6

[GSH*19] Garon M., Sunkavalli K., Hadap S., Carr N., Lalonde J.-F.: Fast spatially-varying indoor lighting estimation. In *Proc. of CVPR* (2019), IEEE, pp. 6908–6917. 2

[GSY*17] Gardner M.-A., Sunkavalli K., Yumer E., Shen X., Gambaretto E., Gagné C., Lalonde J.-F.: Learning to predict indoor illumination from a single image. *ACM Trans. Graph. 36*, 6 (2017), 1–14. 2

[GZB*13] Gkioulekas I., Zhao S., Bala K., Zickler T., Levin A.: Inverse volume rendering with material dictionaries. *ACM Trans. Graph. 32*, 6 (nov 2013), 162:1–162:13. 3

[IKH*11] Izadi S., Kim D., Hilliges O., Molyneaux D., Newcombe R., Kohli P., Shotton J., Hodges S., Freeman D., Davison A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of ACM UIST* (2011), pp. 559–568. 2

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 6

[KCW*18] Kang K., Chen Z., Wang J., Zhou K., Wu H.: Efficient reflectance capture using an autoencoder. *ACM Trans. Graph. 37*, 4 (2018), 127:1–127:10. 2

[KHFH11] Karsch K., Hedau V., Forsyth D., Hoiem D.: Rendering synthetic objects into legacy photographs. *ACM Trans. Graph. 30*, 6 (2011), 1–12. 3

[KKSH17] Kasper M., Keivan N., Sibley G., Heckman C. R.: Light source estimation with analytical path-tracing. *CoRR abs/1701.04101* (2017). 3

[KSH*14] Karsch K., Sunkavalli K., Hadap S., Carr N., Jin H., Fonte R., Sittig M., Forsyth D.: Automatic scene inference for 3d object compositing. *ACM Trans. Graph. 33*, 3 (2014), 1–15. 3

[KUH18] Kato H., Ushiku Y., Harada T.: Neural 3D mesh renderer. In *Proc. of CVPR* (2018), IEEE, pp. 3907–3916. 3

[LADL18] Li T.-M., Aittala M., Durand F., Lehtinen J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. 37*, 6 (2018), 222:1–222:11. 2, 3, 4

[LB14] Loper M. M., Black M. J.: OpenDR: An approximate differentiable renderer. In *ECCV* (sep 2014), vol. 8695, pp. 154–169. 3

[LGZ*20] Liu A., Ginosar S., Zhou T., Efros A. A., Snavely N.: Learning to factorize and relight a city. In *ECCV* (2020). 3

[LHJ19] Loubet G., Holzschuch N., Jakob W.: Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics* (Dec. 2019). 3

[LSR*20] Li Z., Shafiei M., Ramamoorthi R., Sunkavalli K., Chandraker M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proc. of CVPR* (2020), IEEE. 2, 3, 7

[LSSS18] Lombardi S., Saragih J., Simon T., Sheikh Y.: Deep appearance models for face rendering. *ACM Trans. Graph. 37*, 4 (2018), 68. 2

[LXR*18] Li Z., Xu Z., Ramamoorthi R., Sunkavalli K., Chandraker M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph. 37*, 6 (2018), 1–11. 3

[MKC*17] Maier R., Kim K., Cremers D., Kautz J., Niessner M.: Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proc. of CVPR* (2017), IEEE, pp. 3114–3122. 2

[MLTFR19] Maximov M., Leal-Taixe L., Fritz M., Ritschel T.: Deep appearance maps. In *Proc. ICCV* (October 2019). 2

[MMZ*18] Meka A., Maximov M., Zollhoefer M., Chatterjee A., Seidel H.-P., Richardt C., Theobalt C.: Lime: Live intrinsic material estimation. In *Proc. of CVPR* (June 2018), IEEE. 3

[NDSRJ20] Nimier-David M., Speierer S., Ruiz B., Jakob W.: Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph. 39*, 4 (2020), 146:1–146:15. 6, 7, 9

[NDVZJ19] Nimier-David M., Vicini D., Zeltner T., Jakob W.: Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph.* (Dec. 2019). 3, 6, 7

[NIH*11] Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., Kohi P., Shotton J., Hodges S., Fitzgibbon A.: Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR* (2011), pp. 127–136. 2

[NZIS13] Niessner M., Zollhöfer M., Izadi S., Stamminger M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph. 32*, 6 (2013), 1–11. 2

[PJ92] Polyak B. T., Juditsky A. B.: Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization 30*, 4 (1992), 838–855. 7

[PJH16] Pharr M., Jakob W., Humphreys G.: *Physically Based Rendering: From Theory to Implementation (3rd ed.)*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Nov. 2016. 5

[RH01] Ramamoorthi R., Hanrahan P.: A signal-processing framework for inverse rendering. In *SIGGRAPH* (2001), pp. 117–128. 2

[Rup88] Ruppert D.: *Efficient estimations from a slowly convergent Robbins-Monro process*. Tech. rep., Cornell University Operations Research and Industrial Engineering, 1988. 7

[SC20] Sang S., Chandraker M.: Single-shot neural relighting and svbrdf estimation. In *ECCV* (2020), Springer, pp. 85–101. 3

[SDR*20] Schmitt C., Donné S., Riegler G., Koltun V., Geiger A.: On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *Proc. of CVPR* (2020), IEEE, pp. 3493–3503. 3

[SWM*19] Straub J., Whelan T., Ma L., Chen Y., Wijmans E., Green S., Engel J. J., Mur-Artal R., Ren C., Verma S., Clarkson A., Yan M., Budge B., Yan Y., Pan X., Yon J., Zou Y., Leon K., Carter N., Briales J., Gillingham T., Mueggler E., Pesqueira L., Savva M., Batra D., Strasdat H. M., Nardi R. D., Goesele M., Lovegrove S., Newcombe R.: The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019). 2, 4, 6, 7, 8

[TSG19] Tsai C.-Y., Sankaranarayanan A. C., Gkioulekas I.: Beyond volumetric albedo–a surface optimization framework for non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 1545–1555. 3, 5

[TZN19] Thies J., Zollhöfer M., Niessner M.: Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 3

[VG95] Veach E., Guibas L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 419–428. 5

[VSJ21] Vicini D., Speierer S., Jakob W.: Path replay backpropagation: Differentiating light paths using constant memory and linear time. *Transactions on Graphics (Proceedings of SIGGRAPH) 40*, 4 (Aug. 2021), 108:1–108:14. doi:10.1145/3450626.3459804. 9

[XSHR18] Xu Z., Sunkavalli K., Hadap S., Ramamoorthi R.: Deep image-based relighting from optimal sparse samples. *ACM Trans. Graph. 37*, 4 (2018), 126. 2

[YDMH99] Yu Y., Debevec P., Malik J., Hawkins T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH* (1999), pp. 215–224. 3

[ZCC16a] Zhang E., Cohen M. F., Curless B.: Emptying, refurnishing, and relighting indoor spaces. *ACM Trans. Graph. 35*, 6 (2016), 1–14. 3

[ZCC16b] Zhang E., Cohen M. F., Curless B.: Emptying, refurnishing, and relighting indoor spaces. *ACM Trans. Graph. 35*, 6 (2016). 3