Inverse Path Tracing for Joint Material and Lighting Estimation

Dejan Azinović[1]　　　　Tzu-Mao Li[2,3]　　　　Anton Kaplanyan[3]　　　　Matthias Nießner[1]

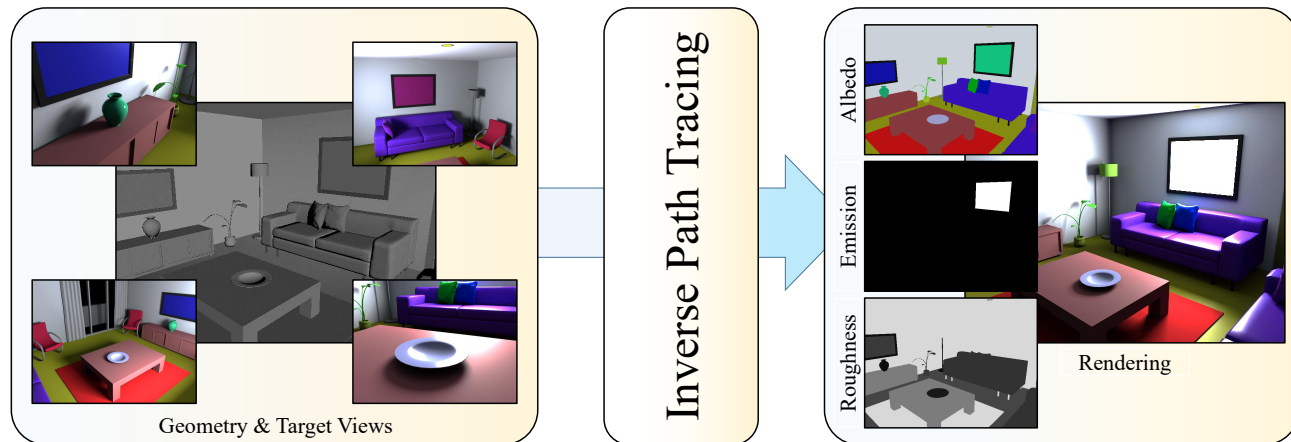[1]Technical University of Munich　　[2]MIT CSAIL　　[3]Facebook Reality Labs

Figure 1: Our Inverse Path Tracing algorithm takes as input a 3D scene and up to several RGB images (left), and estimates material as well as the lighting parameters of the scene. The main contribution of our approach is the formulation of an end-to-end differentiable inverse Monte Carlo renderer which is utilized in a nested stochastic gradient descent optimization.

Abstract

Modern computer vision algorithms have brought significant advancement to 3D geometry reconstruction. However, illumination and material reconstruction remain less studied, with current approaches assuming very simplified models for materials and illumination. We introduce Inverse Path Tracing, a novel approach to jointly estimate the material properties of objects and light sources in indoor scenes by using an invertible light transport simulation. We assume a coarse geometry scan, along with corresponding images and camera poses. The key contribution of this work is an accurate and simultaneous retrieval of light sources and physically based material properties (e.g., diffuse reflectance, specular reflectance, roughness, etc.) for the purpose of editing and re-rendering the scene under new conditions. To this end, we introduce a novel optimization method using a differentiable Monte Carlo renderer that computes derivatives with respect to the estimated unknown illumination and material properties. This enables joint optimization for physically correct light transport and material models using a tailored stochastic gradient descent.

1. Introduction

With the availability of inexpensive, commodity RGB-D sensors, such as the Microsoft Kinect, Google Tango, or Intel RealSense, we have seen incredible advances in 3D reconstruction techniques [27, 14, 28, 33, 8]. While tracking and reconstruction quality have reached impressive levels, the estimation of lighting and materials has often been neglected. Unfortunately, this presents a serious problem for virtual- and mixed-reality applications, where we need to re-render scenes from different viewpoints, place virtual objects, edit scenes, or enable telepresence scenarios where a person is placed in a different room.

This problem has been viewed in the 2D image domain, resulting in a large body of work on intrinsic images or videos [1, 26, 25]. However, the problem is severely underconstrained on monocular RGB data due to lack of known geometry, and thus requires heavy regularization to jointly solve for lighting, material, and scene geometry. We believe that the problem is much more tractable in the context of given 3D reconstructions. However, even with depth data available, most state-of-the-art methods, *e.g.*, shading-based refinement [34, 37] or indoor re-lighting [36], are based on simplistic lighting models, such as spher-

ical harmonics (SH) [30] or spatially-varying SH [23], which can cause issues on occlusion and view-dependent effects (Fig. 4).

In this work, we address this shortcoming by formulating material and lighting estimation as a proper inverse rendering problem. To this end, we propose an Inverse Path Tracing algorithm that takes as input a given 3D scene along with a single or up to several captured RGB frames. The key to our approach is a differentiable Monte Carlo path tracer which can differentiate with respect to rendering parameters constrained on the difference of the rendered image and the target observation. Leveraging these derivatives, we solve for the material and lighting parameters by nesting the Monte Carlo path tracing process into a stochastic gradient descent (SGD) optimization. The main contribution of this work lies in this SGD optimization formulation, which is inspired by recent advances in deep neural networks.
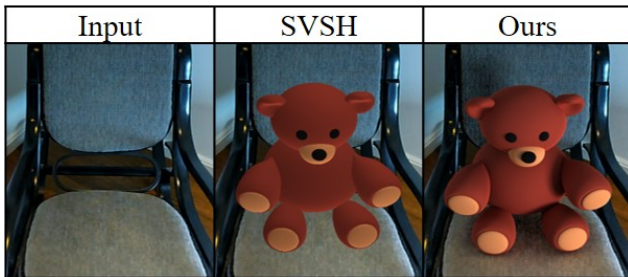

Figure 2: Inserting virtual objects in real 3D scenes; the estimated lighting and material parameters of our approach enable convincing image compositing in AR settings.

We tailor this Inverse Path Tracing algorithm to 3D scenes, where scene geometry is (mostly) given but the material and lighting parameters are unknown. In a series of experiments on both synthetic ground truth and real scan data, we evaluate the design choices of our optimizer. In comparison to state-of-the-art lighting models, we show that our inverse rendering formulation achieves significantly more accurate results.

In summary, we contribute the following:

- An end-to-end differentiable inverse path tracing formulation for joint material and lighting estimation.
- A flexible stochastic optimization framework with extensibility and flexibility for different materials and regularization terms.

2. Related Work

Material and illumination reconstruction has a long history in computer vision (*e.g.*, [29, 4]). Given scene geometry and observed radiance of the surfaces, the task is to infer the material properties and locate the light source. However, to our knowledge, none of the existing methods handle non-Lambertian materials with near-field illumination (area light sources), while taking interreflection between surfaces into account.

3D approaches. A common assumption in reconstructing material and illumination is that the light sources are infinitely far away. Ramamoorthi and Hanrahan [30] project both material and illumination onto spherical harmonics and solve for their coefficients using the convolution theorem. Dong *et al*. [11] solve for spatially-varying reflectance from a video of an object. Kim *et al*. [19] reconstruct the reflectance by training a convolutional neural network operating on voxels constructed from RGB-D video. Maier *et al*. [23] generalize spherical harmonics to handle spatial dependent effects, but do not correctly take view-dependent reflection and occlusion into account. All these approaches simplify the problem by assuming that the light sources are infinitely far away, in order to reconstruct a single environment map shared by all shading points. In contrast, we model the illumination as emission from the surfaces, and handle near-field effects such as the squared distance falloff or glossy reflection better.

Image-space approaches (*e.g.*, [2, 1, 10, 25]). These methods usually 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 geometry factors such as the squared distance falloff in a physically based manner. These methods also usually require a huge amount of training data, and are prone to errors when subjected to scenes with different characteristics from the training data.

Active illumination (*e.g.*, [24, 9, 16]). These methods use highly-controlled lighting for reconstruction, by carefully placing the light sources and measuring the intensity. These methods produce high-quality results, at the cost of a more complicated setup.

Inverse radiosity (*e.g.*, [35, 36]) achieves impressive results for solving near-field illumination and Lambertian materials for indoor illumination. It is difficult to generalize the radiosity algorithm to handle non-Lambertian materials (Yu *et al*. handle it by explicitly measuring the materials, whereas Zhang *et al*. assume Lambertian).

Differentiable rendering. Blanz and Vetter utilized differentiable rendering for face reconstruction using 3D morphable models [3], which is now leveraged by modern analysis-by-synthesis face trackers [31]. Gkioulekas *et al*. [13, 12] and Che *et al*. [7] solve for scattering parameters using a differentiable volumetric path tracer. Kasper *et al*. [17] developed a differentiable path tracer, but focused on distant illumination. Loper and Black [22] and Kato [18] developed fast differentiable rasterizers, but do not support global illumination. Li *et al*. [21] showed that it is possible to compute correct gradients of a path tracer while taking discontinuities introduced by visibility into consideration.

3. Method

Our *Inverse Path Tracing* method employs physically based light transport simulation [15] to estimate derivatives of all unknown parameters w.r.t. the rendered image(s). The rendering problem is generally extremely high-dimensional and is therefore usually solved using stochastic integration methods, such as Monte Carlo integration. In this work, we nest differentiable path tracing into stochastic gradient descent to solve for the unknown scene parameters. Fig. 3 illustrates the workflow of our approach. We start from the captured imagery, scene geometry, object segmentation of the scene, and an arbitrary initial guess of the illumination and material parameters. Material and emission properties are then estimated by optimizing for rendered imagery to match the captured images.

The path tracer renders a noisy and undersampled version of the image using Monte Carlo integration and computes derivatives of each sampled light path w.r.t. the unknowns. These derivatives are passed as input to our optimizer to perform a single optimization step. This process is performed iteratively until we arrive at the correct solution. Path tracing is a computationally expensive operation, and this optimization problem is non-convex and ill-posed. To this end, we employ variance reduction and novel regularization techniques (Sec. 4.4) for our gradient computation to arrive at a converged solution within a reasonable amount of time, usually a few minutes on a modern 8-core CPU.

3.1. Light Transport Simulation

If all scene and image parameters are known, an expected linear pixel intensity can be computed using light transport simulation. In this work, we assume that all surfaces are opaque and there is no participating media (*e.g.*, fog) in the scene. In this case, the rendered intensity I_R^j for pixel j is computed using the path integral [32]:

$$I_R^j = \int_\Omega h_j(\mathbf{X}) f(\mathbf{X}) d\mu(\mathbf{X}), \tag{1}$$

where $\mathbf{X} = (\mathbf{x}_0, ..., \mathbf{x}_k)$ is a light path, *i.e.* a list of vertices on the surfaces of the scene starting at the light source and ending at the sensor; the integral is a path integral taken over the space of all possible light paths of all lengths, denoted as Ω, with a product area measure $\mu(\cdot)$; $f(\mathbf{X})$ is the measurement contribution function of a light path $\mathbf{X}$ that computes how much energy flows through this particular path; and $h_j(\mathbf{X})$ is the pixel filter kernel of the sensor's pixel j, which is non-zero only when the light path $\mathbf{X}$ ends around the pixel j and incorporates sensor sensitivity at this pixel. We refer interested readers to the work of Veach [32] for more details on the light transport path integration.

The most important term of the integrand to our task is the path measurement contribution function f, as it contains

the material parameters as well as the information about the light sources. For a path $\mathbf{X} = (\mathbf{x}_0, ..., \mathbf{x}_k)$ of length k, the measurement contribution function has the following form:

$$f(\mathbf{X}) = L_e(\mathbf{x}_0, \overline{\mathbf{x}_0\mathbf{x}_1}) \prod_{i=1}^k f_r(\mathbf{x}_i, \overline{\mathbf{x}_{i-1}\mathbf{x}_i}, \overline{\mathbf{x}_i\mathbf{x}_{i+1}}), \tag{2}$$

where L_e is the radiance emitted at the scene surface point $\mathbf{x}_0$ (beginning of the light path) towards the direction $\overline{\mathbf{x}_0\mathbf{x}_1}$. At every interaction vertex $\mathbf{x}_i$ of the light path, there is a *bidirectional reflectance distribution function* (BRDF) $f_r(\mathbf{x}_i, \overline{\mathbf{x}_{i-1}\mathbf{x}_i}, \overline{\mathbf{x}_i\mathbf{x}_{i+1}})$ defined. The BRDF describes the material properties at the point $\mathbf{x}_i$, *i.e.*, how much light is scattered from the incident direction $\overline{\mathbf{x}_{i-1}\mathbf{x}_i}$ towards the outgoing direction $\overline{\mathbf{x}_i\mathbf{x}_{i+1}}$. The choice of the parametric BRDF model f_r is crucial to the range of materials that can be reconstructed by our system. We discuss the challenges of selecting the BRDF model in Sec. 4.1.

Note that both the BRDF f_r and the emitted radiance L_e are unknown and the desired parameters to be found at every point on the scene manifold.

3.2. Optimizing for Illumination and Materials

We take as input a series of images in the form of real-world photographs or synthetic renderings, together with the reconstructed scene geometry and corresponding camera poses. We aim to solve for the unknown material parameters $\mathcal{M}$ and lighting parameters $\mathcal{L}$ that will produce rendered images of the scene that are identical to the input images.

Given the un-tonemapped captured pixel intensities I_C^j at all pixels j of all images, and the corresponding noisy estimated pixel intensities $\tilde{I}_R^j$ (in linear color space), we seek all material and illumination parameters $\Theta = \{\mathcal{M}, \mathcal{L}\}$ by solving the following optimization problem using stochastic gradient descent:

$$\underset{\Theta}{\operatorname{argmin}} E(\Theta) = \sum_j^N \left| I_C^j - \tilde{I}_R^j \right|_1, \tag{3}$$

where N is the number of pixels in all images. We found that using an L_1 norm as a loss function helps with robustness to outliers, such as extremely high contribution samples coming from Monte Carlo sampling.

3.3. Computing Gradients with Path Tracing

In order to efficiently solve the minimization problem in Eq. 3 using stochastic optimization, we compute the gradient of the energy function $E(\Theta)$ with respect to the set of unknown material and emission parameters Θ:

$$\nabla_\Theta E(\Theta) = \sum_j^N \nabla_\Theta \tilde{I}_R^j \operatorname{sgn}\left(I_C^j - \tilde{I}_R^j \right), \tag{4}$$

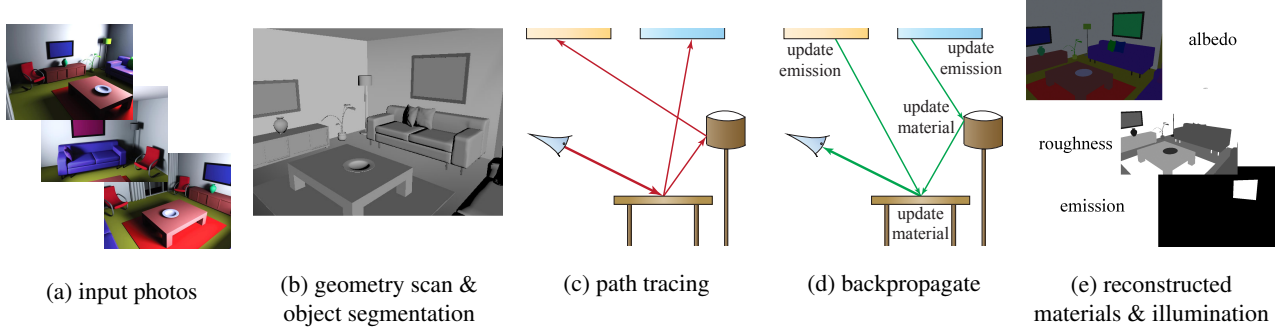| (a) input photos | (b) geometry scan & object segmentation | (c) path tracing | (d) backpropagate | (e) reconstructed materials & illumination |

Figure 3: Overview of our pipeline. Given (a) a set of input photos from different views, along with (b) an accurate geometry scan and proper segmentation, we reconstruct the material properties and illumination of the scene, by iteratively (c) rendering the scene with path tracing, and (d) backpropagating to the material and illumination parameters in order to update them. After numerous iterations, we obtain the (e) reconstructed material and illumination.

where $\mathrm{sgn}(\cdot)$ is the sign function, and $\nabla_\Theta \tilde{I}_R^j$ the gradient of the Monte Carlo estimate with respect to all unknowns Θ.

Note that this equation for computing the gradient now has two Monte Carlo estimates for each pixel j: (1) the estimate of pixel color itself $\tilde{I}_R^j$; and (2) the estimate of its gradient $\nabla_\Theta \tilde{I}_R^j$. Since the expectation of product only equals the product of expectation when the random variables are independent, it is important to draw independent samples for each of these estimates to avoid introducing bias.

In order to compute the gradients of a Monte Carlo estimate for a single pixel j, we determine what unknowns are touched by the measurement contribution function $f(\mathbf{X})$ for a sampled light path $\mathbf{X}$. We obtain the explicit formula of the gradients by differentiating Eq. 2 using the product rule (for brevity, we omit some arguments for emission L_e and BRDF f_r):

$$\nabla_{\Theta_{\mathcal{L}}} f(\mathbf{X}) = \nabla_{\Theta_{\mathcal{L}}} L_e(\mathbf{x}_0) \prod_i^k f_r(\mathbf{x}_i) \qquad (5)$$

$$\nabla_{\Theta_{\mathcal{M}}} f(\mathbf{X}) = L_e(\mathbf{x}_0) \sum_l^k \nabla_{\Theta_{\mathcal{M}}} f_r(\mathbf{x}_l) \prod_{i,i\neq l}^k f_r(\mathbf{x}_i) \qquad (6)$$

where the gradient vector $\nabla_\Theta = \{\nabla_{\Theta_{\mathcal{M}}}, \nabla_{\Theta_{\mathcal{L}}}\}$ is very sparse and has non-zero values only for unknowns touched by the path $\mathbf{X}$. The gradients of emissions (Eq. 5) and materials (Eq. 6) have similar structure to the original path contribution (Eq. 2). Therefore, it is natural to apply the same path sampling strategy; see the appendix for details.

3.4. Multiple Captured Images

The single-image problem can be directly extended to multiple images. Given multiple views of a scene, we aim to find parameters for which rendered images from these views match the input images. A set of multiple views can cover parts of the scene that are not covered by any single

view from the set. This proves important for deducing the correct position of the light source in the scene. With many views, the method can better handle view-dependent effects such as specular and glossy highlights, which can be ill-posed with just a single view, as they can also be explained as variations of albedo texture.

4. Optimization Parameters and Methodology

In this section we address the remaining challenges of the optimization task: what are the material and illumination parameters we actually optimize for, and how to resolve the ill-posed nature of the problem.

4.1. Parametric Material Model

We want our material model to satisfy several properties. First, it should cover as much variability in appearance as possible, including such common effects as specular highlights, multi-layered materials, and spatially-varying textures. On the other hand, since each parameter adds another unknown to the optimization, we would like to keep the number of parameters minimal. Since we are interested in re-rendering and related tasks, the material model needs to have interpretable parameters, so the users can adjust the parameters to achieve the desired appearance. Finally, since we are optimizing the material properties using first-order gradient-based optimization, we would like the range of the material parameters to be similar.

To satisfy these properties, we represent our materials using the Disney material model [5], the state-of-the-art physically based material model used in movie and game rendering. It has a "base color" parameter which is used by both diffuse and specular reflectance, as well as 10 other parameters describing the roughness, anisotropy, and specularity of the material. All these parameters are perceptually mapped to $[0, 1]$, which is both interpretable and suitable for optimization.
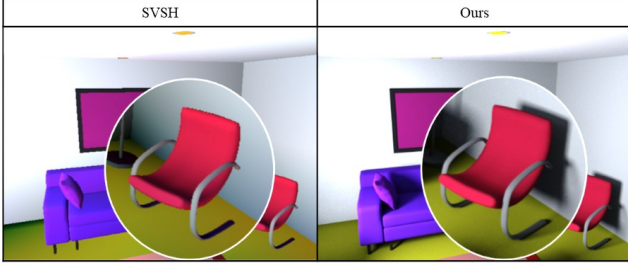
SVSH	Ours

Figure 4: Methods based on spherical harmonics have difficulties handling sharp shadows or lighting changes due to the distant illumination assumption. A physically based method, such as Inverse Path Tracing, correctly reproduces these effects.

4.2. Scene Parameterization

We use triangle meshes to represent the scene geometry. Surface normals are defined per-vertex and interpolated within each triangle using barycentric coordinates. The optimization is performed on a per-object basis, *i.e.*, every object has a single unknown emission and a set of material parameters that are assumed constant across the whole object. We show that this is enough to obtain accurate lighting and an average constant value for the albedo of an object.

4.3. Emission Parameterization

For emission reconstruction, we currently assume all light sources are scene surfaces with an existing reconstructed geometry. For each emissive surface, we currently assume that emitted radiance is distributed according to a view-independent directional emission profile $L_e(\mathbf{x}, \mathbf{i}) = e(\mathbf{x})(\mathbf{i} \cdot \mathbf{n}(\mathbf{x}))_+$, where $e(\mathbf{x})$ is the unknown radiant flux at $\mathbf{x}$; $\mathbf{i}$ is the emission direction at surface point $\mathbf{x}$, $\mathbf{n}(\mathbf{x})$ is the surface normal at $\mathbf{x}$ and $(\cdot)_+$ is the dot product (cosine) clamped to only positive values. This is a common emission profile for most of the area lights, which approximates most of the real soft interior lighting well. Our method can also be extended to more complex or even unknown directional emission profiles or purely directional distant illumination (*e.g.*, sky dome, sun) if needed.

4.4. Regularization

The observed color of an object in a scene is most easily explained by assigning emission to the triangle. This is only avoided by differences in shading of the different parts of the object. However, it can happen that there are no observable differences in the shading of an object, especially if the object covers only a few pixels in the input image. This can be a source of error during optimization. Another source of error is Monte Carlo and SGD noise. These errors lead to incorrect emission parameters for many objects after the optimization. The objects usually have a small estimated emission value when they should have none. We tackle the

problem with an L1-regularizer for the emission. The vast majority of objects in the scene is not an emitter and having such a regularizer suppresses the small errors we get for the emission parameters after optimization.

4.5. Optimization Parameters

We use ADAM [20] as our optimizer with batch size $B = 8$ estimated pixels and learning rate $5 \cdot 10^{-3}$. To form a batch, we sample B pixels uniformly from the set of all pixels of all images. Please see the appendix for an evaluation regarding the impact of different batch sizes and sampling distributions on the convergence rate. While a higher batch size reduces the variance of each iteration, having smaller batch sizes, and therefore faster iterations, proves to be more beneficial.

5. Results

Evaluation on synthetic data. We first evaluate our method on multiple synthetic scenes, where we know the ground truth solution. Quantitative results are listed in Tab. 1, and qualitative results are shown in Fig. 5. Each scene is rendered using a path tracer with the ground truth lighting and materials to obtain the "captured images". These captured images and scene geometry are then given to our Inverse Path Tracing algorithm, which optimizes for unknown lighting and material parameters. We compare to the closest previous work based on spatially-varying spherical harmonics (SVSH) [23]. SVSH fails to capture sharp details such as shadows or high-frequency lighting changes. A comparison of the shadow quality is presented in Fig. 4.

Our method correctly detects light sources and converges to a correct emission value, while the emission of objects that do not emit light stays at zero. Fig. 6 shows a novel view, rendered with results from an optimization that was performed on input views from Fig. 5. Even though the light source was not visible in any of the input views, its emission was correctly computed by Inverse Path Tracing.

In addition to albedo, our Inverse Path Tracer can also optimize for other material parameters such as roughness. In Fig. 8, we render a scene containing objects of varying roughness. Even when presented with the challenge of estimating both albedo and roughness, our method produces the correct result as shown in the re-rendered image.

Evaluation on real data. We use the Matterport3D [6] dataset to evaluate our method on real captured scenes obtained through 3D reconstruction. The scene was parameterized using the segmentation provided in the dataset. Due to imperfections in the data, such as missing geometry and inaccurate surface normals, it is more challenging to perform an accurate light transport simulation. Nevertheless, our method produces impressive results for the given input. After the optimization, the optimized light direction
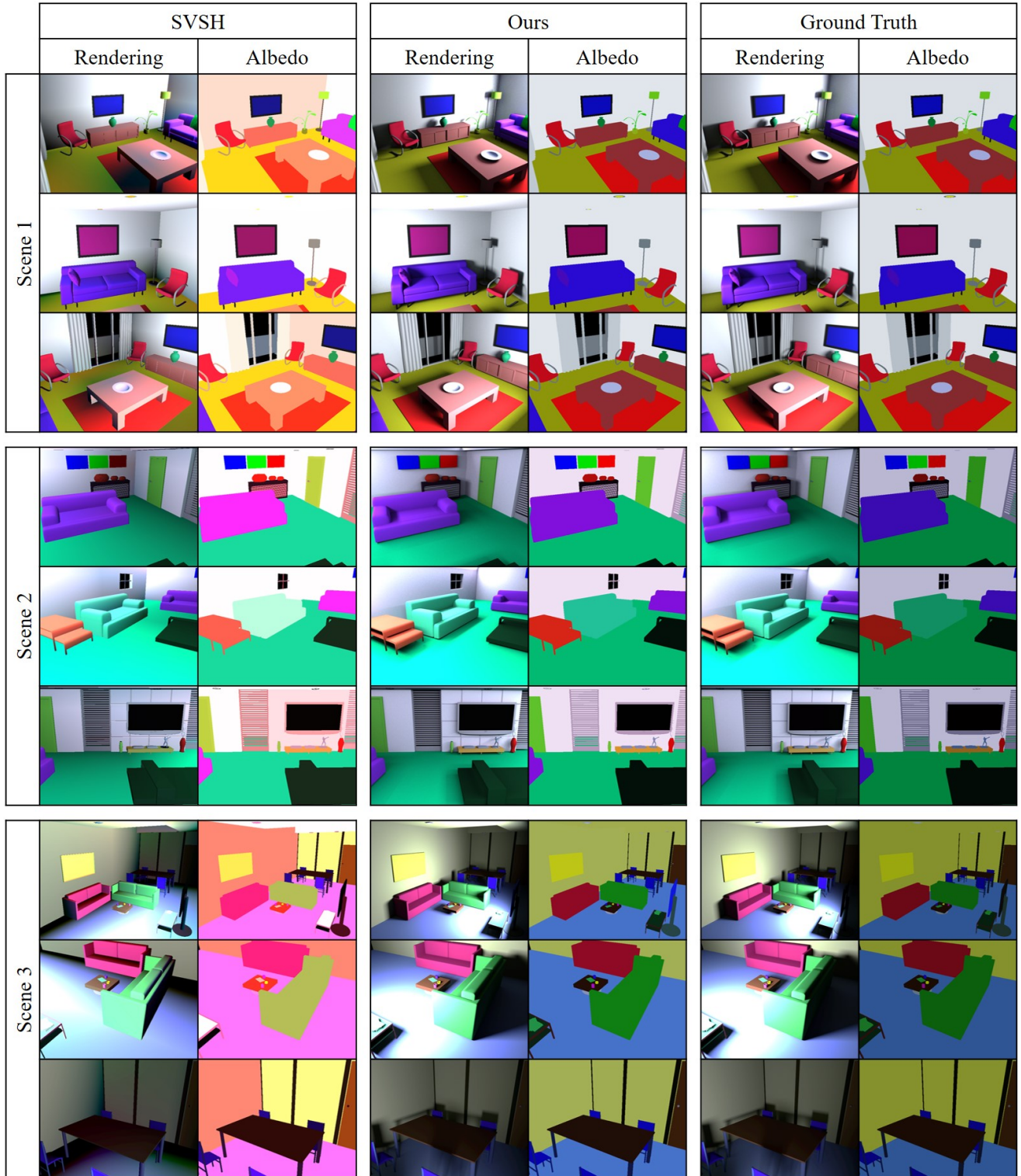
Figure 5: Evaluation on synthetic scenes. Three scenes have been rendered from different views with both direct and indirect lighting (right). An approximation of the albedo lighting with spatially-varying spherical harmonics is shown (left). Our method is able to detect the light source even though it was not observed in any of the views (middle). Notice that we are able to reproduce sharp lighting changes and shadows correctly. The albedo is also closer to the ground truth albedo.
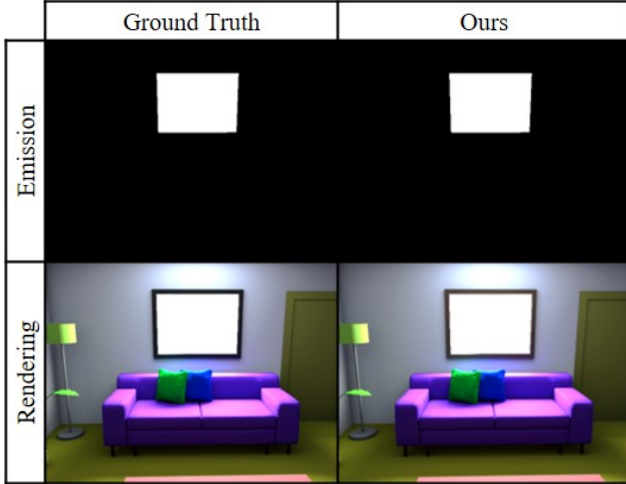
Figure 6: Inverse Path Tracing is able to correctly detect the light emitting object (top). The ground truth rendering and our estimate is shown on the bottom. Note that this view was not used during optimization.
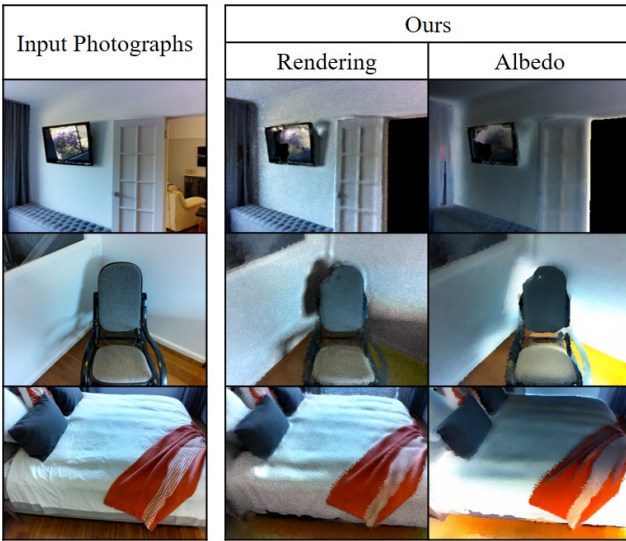

Figure 7: We can resolve object textures by optimizing for the unknown parameters per triangle. Higher resolution textures can be obtained by further subdividing the geometry.
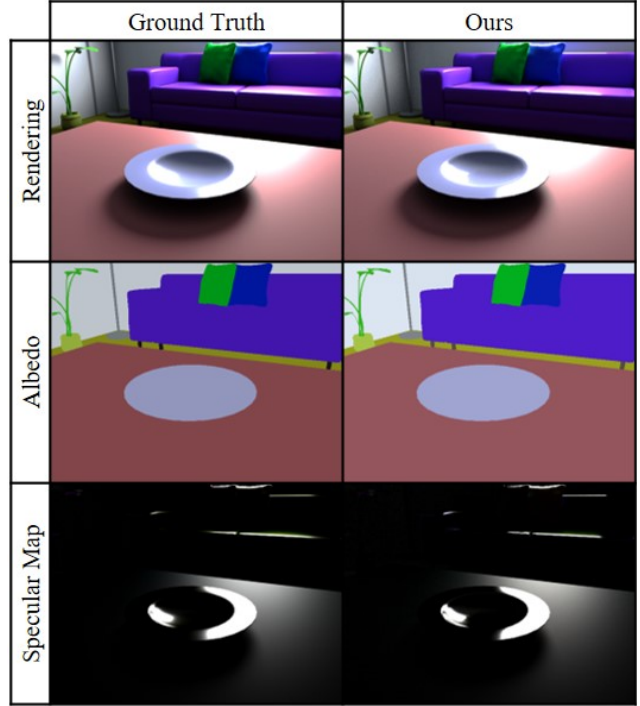

Figure 8: Inverse Path Tracing is agnostic to the underlying BRDF; *e.g.*, here, in a specular case, we are able to correctly estimate both the albedo and the roughness of the objects. The ground truth rendering and our estimate is shown on top, the albedo in the middle and the specular map on the bottom.

Optimizer Ablation. There are several ways to reduce the variance of our optimizer. One obvious way is to use more samples to estimate the pixel color and the derivatives, but this also results in slower iterations. Fig. 9 shows that the method does not converge if only a single path is used. A general recommendation is to use between 2^7 and 2^{10} depending on the scene complexity and number of unknowns.

Another important aspect of our optimizer is the sample distribution for pixel color and derivatives estimation. Our tests in Fig. 10 show that minimal variance can be achieved by using one sample to estimate the derivatives and the remaining samples in the available computational budget to estimate the pixel color.

Limitations. Inverse Path Tracing assumes that high-quality geometry is available. However, imperfections in the recovered geometry can have big impact on the quality of material estimation as shown in Fig. 11. Our method also does not compensate for the distortions in the captured input images. Most cameras, however, produce artifacts such as lens flare, motion blur or radial distortion. Our method can potentially account for these imperfections by simulating the corresponding effects and optimize not only for the ma-

matches the captured light direction and the rendered result closely matches the photograph. Fig. 11 shows a comparison to the SVSH method.

The albedo of real-world objects varies across its surface. Inverse Path Tracing is able to compute an object's average albedo by employing knowledge of the scene segmentation. To reproduce fine texture, we refine the method to optimize for each individual triangle of the scene with adaptive subdivision where necessary. This is demonstrated in Fig. 7.
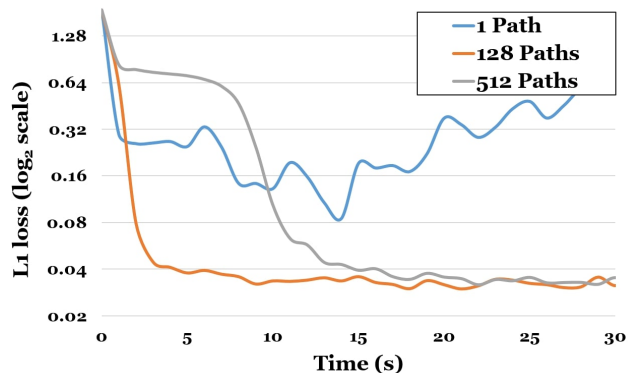
Figure 9: Convergence with respect to the number of paths used to estimate the pixel color. If this is set too low, the algorithm will fail.
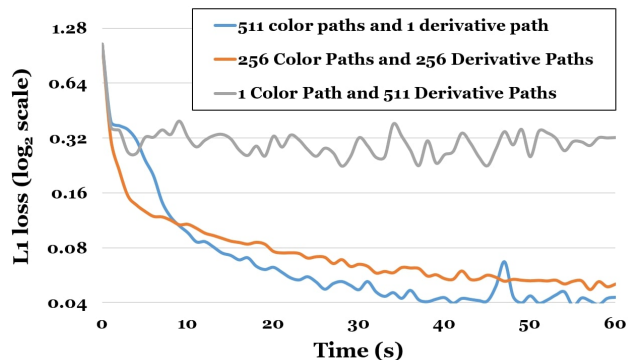

Figure 10: Convergence with respect to distributing the available path samples budget between pixel color and derivatives. It is best to keep the number of paths high for pixel color estimation and low for derivative estimation.

Method	Scene 1	Scene 2	Scene 3
SVSH Rendering Loss	0.052	0.048	0.093
Our Rendering Loss	**0.006**	**0.010**	**0.003**
SVSH Albedo Loss	0.052	0.037	0.048
Our Albedo Loss	**0.002**	**0.009**	**0.010**

Table 1: Quantitative evaluation for synthetic data. We measure the L1 loss with respect to the rendering error and the estimated albedo parameters. Note that our approach achieves a significantly lower error on both metrics.

terial parameters, but also for the camera parameters, which we leave for future work.

6. Conclusion

We present Inverse Path Tracing, a novel approach for joint lighting and material estimation in 3D scenes. We
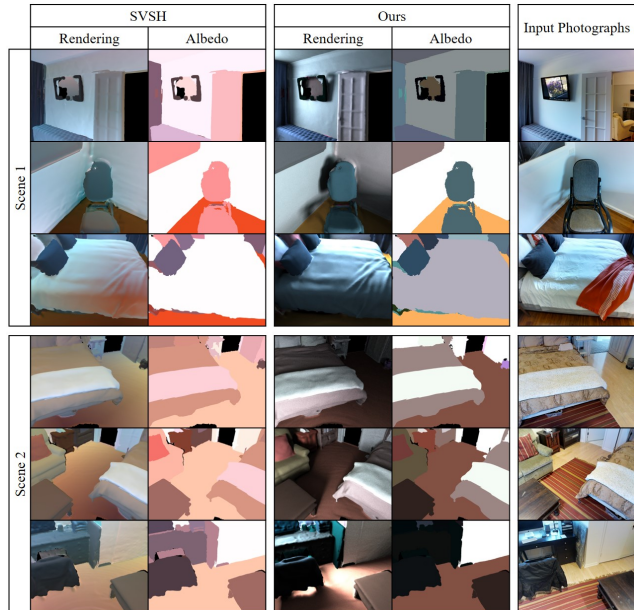

Figure 11: Evaluation on real scenes: (right) input is 3D scanned geometry and photographs. We employ object instance segmentation to estimate the emission and the average albedo of every object in the scene. Our method is able to optimize for the illumination and shadows. Other methods usually do not take occlusions into account and fail to model shadows correctly. Views 1 and 2 of Scene 2 show that if the light emitters are not present in the input geometry, our method gives an incorrect estimation.

demonstrate that our differentiable Monte Carlo renderer can be efficiently integrated in a nested stochastic gradient descent optimization. In our results, we achieve significantly higher accuracy than existing approaches. High-fidelity reconstruction of materials and illumination is an important step for a wide range of applications such as virtual and augmented reality scenarios. Overall, we believe that this is a flexible optimization framework for computer vision that is extensible to various scenarios, noise factors, and other imperfections of the computer vision pipeline. We hope to inspire future work along these lines, for instance, by incorporating more complex BRDF models, joint geometric refinement and completion, and further stochastic regularizations and variance reduction techniques.

Acknowledgements

References

[1] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, 2015. 1, 2

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

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999. 2

[4] Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. Intrinsic decompositions for image editing. *Computer Graphics Forum (Eurographics State of the Art Reports)*, 36(2), 2017. 2

[5] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. 4

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 5

[7] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Inverse transport networks. *arXiv preprint arXiv:1809.10820*, 2018. 2

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017. 1

[9] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. SIGGRAPH, pages 145–156, 2000. 2

[10] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37(4):128:1–128:15, 2018. 2

[11] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 33(6):193:1–193:12, 2014. 2

[12] Ioannis Gkioulekas, Anat Levin, and Todd Zickler. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision*, pages 685–701, 2016. 2

[13] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.*, 32(6):162:1–162:13, nov 2013. 2

[14] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 1

[15] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, Aug. 1986. 3

[16] Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. Efficient reflectance capture using an autoencoder. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37(4):127:1–127:10, 2018. 2

[17] Mike Kasper, Nima Keivan, Gabe Sibley, and Christoffer R. Heckman. Light source estimation with analytical path-tracing. *CoRR*, abs/1701.04101, 2017. 2

[18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 2

[19] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Niessner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation, Oct 2017. 2

[20] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[21] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 2

[22] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision*, volume 8695, pages 154–169, sep 2014. 2

[23] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3d: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017. 2, 5

[24] Stephen Robert Marschner. *Inverse Rendering for Computer Graphics*. PhD thesis, 1998. 2

[25] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[26] Abhimitra Meka, Michael Zollhoefer, Christian Richardt, and Christian Theobalt. Live intrinsic video. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 35(4), 2016. 1

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 1

[28] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013. 1

[29] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. *Computer Graphics Forum*, 22(4):663–687, 2003. 2

[30] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. SIGGRAPH, pages 117–128, 2001. 2

[31] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 2

[32] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162. 3

[33] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 1

[34] Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(6), 2014. 1

[35] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH*, pages 215–224, 1999. 2

[36] Edward Zhang, Michael F. Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016. 1, 2

[37] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG)*, 34(4), 2015. 1