UNIVERSITY OF CALIFORNIA SAN DIEGO

Physically-Motivated Learning For Photorealistic Scene Reconstruction and Editing in the Wild

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Zhengqin Li

Committee in charge:

Professor Manmohan Chandraker, Chair
Professor David Kriegman
Professor Ravi Ramamoorthi
Professor Lawrence Saul
Professor Nuno Vasconcelos

2022

The Dissertation of Zhengqin Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my Ph.D. advisor, Manmohan Chandraker. It is a great honor to be one of his first Ph.D. students. His enthusiasm for solving the most fundamental problems, focusing on the big picture, elegant academic writing and presentations, and incredible patience when mentoring students, not only contribute tremendously to my Ph.D. career but also will keep inspiring me to become a better researcher in the future.

Second, I would like to thank my thesis committee, David Kriegman, Ravi Ramamoorthi, Lawrence Saul, Nuno Vasconcelos, for offering helpful feedback.

I greatly benefit from collaborations with my mentors. Kalyan Sunkavalli introduced me to the topic of spatially-varying material reconstruction and participated in nearly every project. His wise suggestions and practical thinking helped me simplify my solutions and avoid messy details whenever I got stuck. Ravi Ramamoorthi and Miloš Hašan generously shared their expertise on computer graphics, especially on rendering, which helped me understand the physics of the image formation process. Zhao Dong shared his research experience in both academia and industry, which helped me make important career decisions.

I would like to acknowledge the contributions of my student collaborators, including Zexiang Xu, Yu-ying Yeh, Sai Bi, Zak Murez, Mohammad Shafiei, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Hong-Xing Yu, and Jonathan Dyssel Stets. They created an encouraging research environment at UC San Diego. I enjoyed working with them and learned a lot from our collaborations.

During my Ph.D., I met many amazing friends who made my graduate life enjoyable, including Sai Bi, Zexiang Xu, Weilun Sun, Lifan Wu, Tiancheng Sun, Alexandr Kuznetsov, Ronald Yu, Meng Song, Yu-ying Yeh, Yao Qin, Mengting Wan, Ting-wei Yu, Shen Sang, Tarun Kalluri, Rui Zhu, Kunal Gupta, Ishit Mehta and so on. Zexiang and Sai are also my long-term collaborators whose insightful suggestions and hands-on experiences helped me significantly in many projects. Weilun taught me important graphics concepts when we worked out at the school gym. Special thanks to Jingwen Wang.

In addition, I would like to thank my undergraduate advisors. I am very grateful to Jiansheng Chen, who introduced me to his lab when I was a second-year student knowing nothing about research. With his guidance, I published my first paper at a top-tier conference and made up my mind to pursue a Ph.D. degree. I was fortunate to work with Olga Veksler remotely as a summer intern at the end of my third year. Her graph-cut work has always been one of my favorite papers in computer vision, motivating me to do research that is practically useful and mathematically interesting.

Finally, I would like to thank my family for unconditionally supporting me to explore my interests.

Chapter 3 is based on the material as it appears in European Conference on Computer Vision (ECCV), 2018 ("Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image", Zhengqin Li, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigator and author of this paper.

Chapter 4 is based on the material as it appears in ACM Transactions on Graphics, 2018 ("Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image", Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigator and author of this paper.

Chapter 5 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020 ("Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes", Zhengqin Li*, Yu-ying Yeh*, Manmohan Chandraker). The dissertation author was the one of the primary investigators and author of this paper. Yu-ying Yeh is a co-first author who contributed equally to the paper.

Chapter 6 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021 ("OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets", Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, Manmohan Chandraker). The dissertation

author was the primary investigators and author of this paper.

Chapter 7 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020 (“Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image”, Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigators and author of this paper.

Chapter 8 is based on the material currently under submission (“Physically-Based Editing of Indoor Scene Lighting from a Single Image”, Zhengqin Li1 Jia Shi, Sai Bi, Rui Zhu, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, Manmohan Chandraker). The dissertation author was the primary investigators and author of this paper.

VITA

2012–2016 Bachelor, Tsinghua University

2016–2022 Doctor of Philosophy, University of California, San Diego

ABSTRACT OF THE DISSERTATION

Physically-Motivated Learning For Photorealistic Scene Reconstruction and Editing in the Wild

by

Zhengqin Li

Doctor of Philosophy in Computer Science

University of California San Diego, 2022

Professor Manmohan Chandraker, Chair

Rapid advances in imaging have made high-quality devices such as mobile phone cameras easily accessible, opening the doors to new applications in image editing and augmentation. They may allow an interior designer to visualize how a kitchen counter will appear after remodeling, or a consumer to see whether a fabric or leather sofa looks better in a living room with color bleeding from walls of various shades, or a real estate agent to demonstrate how a room imaged under fluorescent lights at night will appear in the glow of a sunrise when a window is opened.

Achieving a high degree of photorealism in such applications remains extremely challenging in computer vision and graphics. They require a comprehensive understanding of all the constituent factors of image formation — shape, material and lighting — which exhibit a

wide spectrum of variations and interact in complex ways to create effects such as highlights, shadows and interreflections. Reconstruction of these intrinsic scene components or the ability to edit them is consequently an extremely ill-posed problem and especially so when only a single or a few images are available. Classical measurement-based methods need expensive, carefully calibrated setups. Prior model-based methods assume simplified physical models that break down in the face of diverse real-world appearances. Thus, a learning paradigm merits consideration, but even powerful deep learning methods suffer from a lack of generalization due to the diversity, long-range interactions and paucity of ground truth data associated with complex light transport.

The key insight of this thesis is to develop **physically-motivated learning**, which incorporates the inductive bias of image formation to enable deep neural networks to reason about shape, material and lighting in complex scenes. The success of our approach rests on three advances. First, we develop neural differentiable rendering modules that model the full physics of image formation, including non-local light transport effects such as shadows, interreflections or refraction. Second, we devise physically-valid representations of material and light sources that are compact enough to make learning tractable, yet expressive enough to model realistic appearance such as spatially-varying reflectance or high-frequency specular highlights and light shafts through an open window. Third, we exploit domain knowledge to create large-scale photorealistic synthetic datasets which circumvent the difficulty of obtaining ground truth for spatially-varying material and complex light paths that enable physically-motivated learning to generalize well to real scenes. We demonstrate the success of our approach through results that surpass the state-of-the-art or solve longstanding open challenges in reconstruction and editing of shape, material and lighting in the presence of complex light transport in unconstrained scenes, with just a single image as input.

This dissertation also democratizes research in vision and graphics through open frameworks that allow creation of high-quality virtual environments. Indeed, a key practical impact is to allow users to create realistic visual effects with only a few images captured with a mobile

phone camera. Our high-quality predicted geometry, spatially-varying lighting and materials enable several augmented reality (AR) applications at an unprecedented level of photorealism — including virtual object insertion and material replacement with realistic shadows and color bleeding, transparent shape reconstruction and light source editing (such as turning off lamps or opening windows) with consistent non-local shadows, interreflections and highlights.

Chapter 1

Introduction

1.1 Motivation

Image appearance is governed by complex interactions between the intrinsic components that constitute a three-dimensional (3D) scene. The light emitted by light sources such as lamps or the sun gets reflected or refracted by object surfaces composed of complex materials an indefinite number of times, before reaching the observer to produce the sense of color. The goal of this thesis is to attain a comprehensive understanding of shape, material and lighting in complex 3D scenes by effectively incorporating inductive biases from image formation. Such a decomposition of images, termed **scene reconstruction** or **inverse rendering**, has remained a canonical challenge in computer vision and computer graphics for over five decades. Indoor scenes present one of the hardest settings for this challenge due to the presence of complex visual effects such as specular highlights, long-range interreflections, directional lighting, shadows and objects in arbitrary layouts composed of myriad spatially-varying materials. This thesis achieves significant advances towards solving this classical challenge in computer vision and graphics, by bringing interdisciplinary insights from geometry, physics, and machine learning to develop methods for scene reconstruction in unconstrained environments.

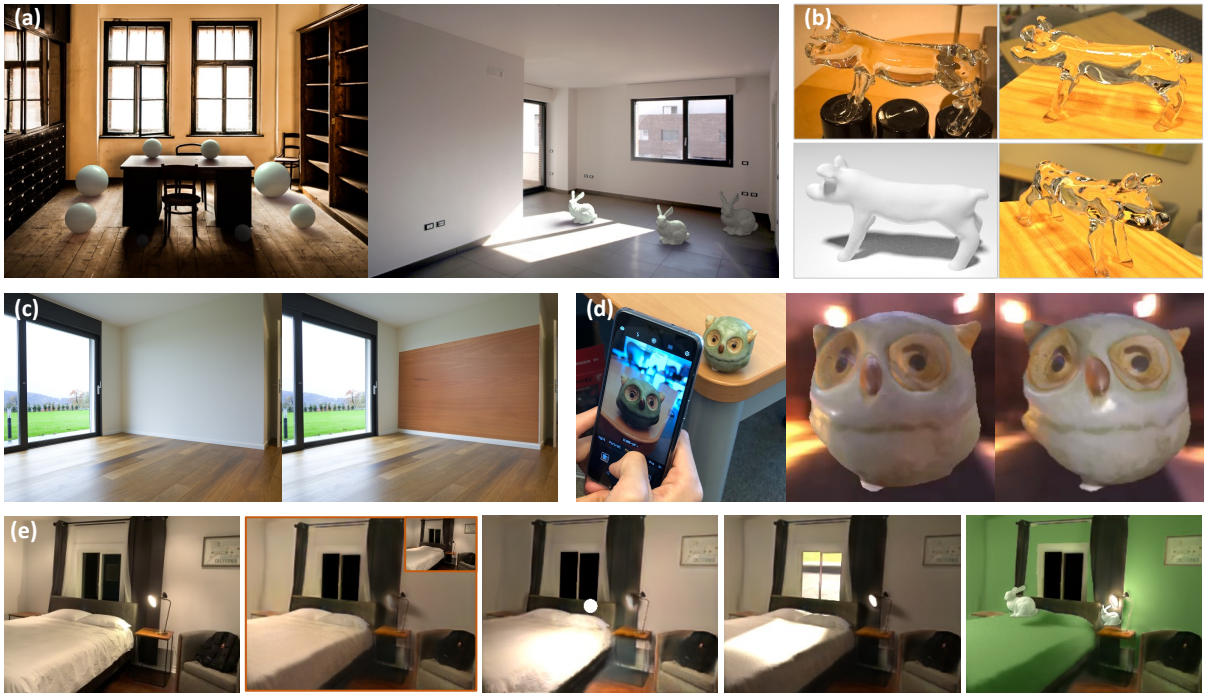


Figure 1.1. Our physically-motivated deep learning enables photorealistic scene editing in unconstrained images captured with a mobile phone camera, such as (a) single-image virtual object insertion, (b) transparent shape relighting and view synthesis using 5–20 images, (c) specular material editing and (d) novel view synthesis and relighting from a single image, (e) various single-image light source editing effects, including turning off an indoor lamp, inserting a virtual lamp, opening a window, or changing wall color with consistent color bleeding. The inset in the second column of (e) shows the ground-truth after turning off the lamp, whose appearance is closely matched by the neural rendering approaches proposed by this thesis. All these effects require a comprehensive understanding of shape, material and lighting, as well as the complex light transport effects manifested in indoor scenes.

1.1.1 Practical Impact

The rapid progress of mobile phone cameras has opened new potential avenues for developing high-quality photorealistic applications for mobile devices. Compared to prior methods that require expensive, calibrated setups or use a simplified image formation model, our frameworks can recover realistic scene properties from even a single image captured by a mobile phone camera in an unconstrained environment. Therefore, it may enable various interesting applications that allow non-expert users to create realistic visual effects, such as those shown in Figure 1.1. This may allow users to insert virtual furniture into their homes or

virtually try-on different wall and floor materials with consistent color-bleeding and shadows rendered realistically, or enable vendors selling products online to create realistic 3D models that customers can visualize under different lighting and from different viewpoints, or allow game developers to capture realistic materials and render them under novel conditions.

In addition, the rising popularity of augmented reality (AR) and virtual reality (VR) creates new requirements for appearance capture. That is, instead of recovering intrinsic properties of a single object or character and rendering it realistically, users may now want to interact with or edit an entire scene – such editing requires that the entire scene appearance be captured. These new requirements may defy classical measurement-based methods, which usually need controlled environments, as well as model-based methods, whose handcrafted priors may not be powerful enough to model the entire scene. In contrast, the frameworks proposed by this thesis can accurately reason about spatially-varying lighting, materials, geometry, and their complex interactions in arbitrary indoor scenes, using only a single image captured in the wild as input. We demonstrate their impact through realistic indoor scene editing applications that cannot be accomplished by any prior works, such as those demonstrated in Figure 1.1 (e). Given the a single real image (first column), we can turn off a lamp (second column) or insert a virtual lamp into the scene (third column) with consistent highlights and shadows. We can also open a virtual window to let high-frequency directional sunlight come into the room (fourth column) or change the wall color with non-local color bleeding being rendered realistically, as shown on the inserted virtual bunnies (fifth column).

In summary, the work described in this thesis will have significant practical impact by democratizing inverse rendering to allow non-experts to easily create, edit, visualize and share their content. It will thereby constitute the foundation for a next generation of AR and VR techniques by enabling the creation of interactive virtual worlds at an unprecedented level of photorealism. The accompanying video [4] summarizes various photorealistic AR applications enabled by our research.

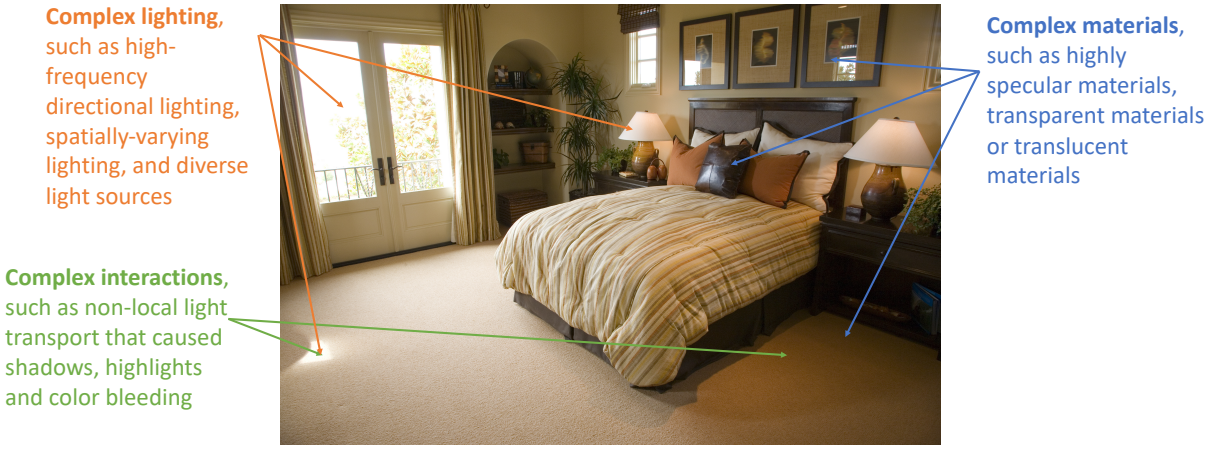


Figure 1.2. The presence of spatially-varying materials, lighting and their complex interactions makes inverse rendering an extremely challenging and ill-posed problem.

1.1.2 Fundamental Challenges

In this section, we further explain the fundamental challenges of scene reconstruction with complex materials and lighting in the wild, which are exemplified in Figure 1.2.

Highly ill-posed

A challenge that immediately manifests is that inverse rendering is a highly ill-posed problem, especially when the observed inputs are sparse. This is because image formation involves infinite-dimensional data, namely spatially-varying lighting, materials and geometry, while its output is just a 3-dimensional RGB color. Therefore, it is extremely difficult to accurately disambiguate the constituent scene factors. Figure 1.3 shows an example where two sets of intrinsic components can both explain the appearance of the input image equally well. The first row consists of the ground truth shape, material and lighting, while the second row bakes the image intensity completely into the diffuse albedo with flat geometry and lighting. However, only the correct intrinsic components reconstructed in the first row can be used to achieve photorealistic editing effects such as inserting a virtual sphere into the scene with consistent shadows and highlights. Besides the dimensionality, a crucial challenge in scene reconstruction with sparse inputs is that constituent factors are often invisible or confounded, yet contribute

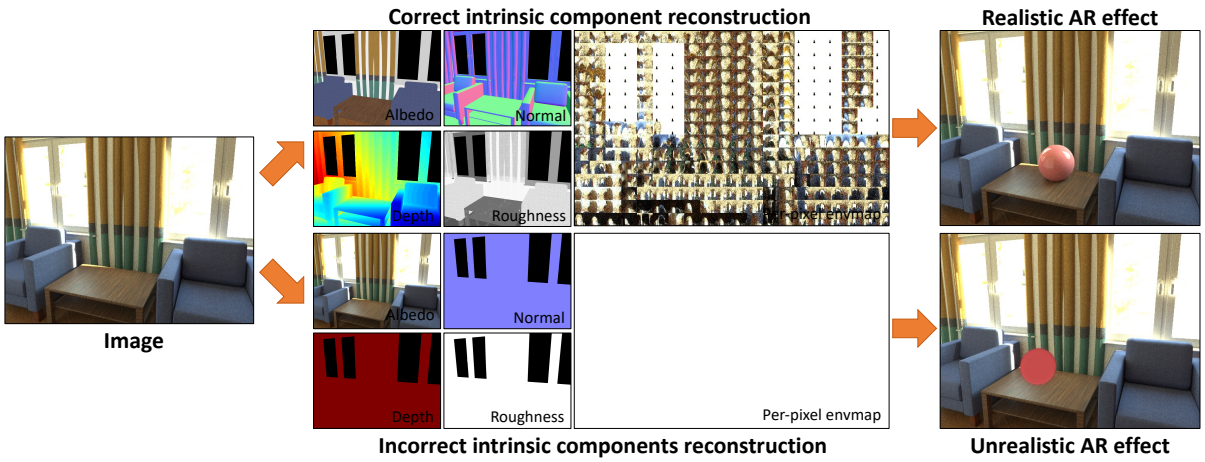


Figure 1.3. An illustration of the ill-posed nature of inverse rendering. While the intrinsic components reconstructed in the first and second row can explain the appearance of the image, only the correct inverse rendering results in the first row can generate realistic AR effects (the shining sphere with realistic highlights and shadows).

significantly to image intensity – such as a ceiling lamp behind the camera or interreflections from an occluded wall.

Complex materials

Real-world material appearance is diverse and complicated. It may exhibit strong high-frequency signals in the angular domain, causing sharp specular highlights with various patterns. A brute-force way to capture material appearance would be to densely sample every view and lighting direction and measure the reflected intensity [132]. While this measurement-based method is effective in a dark room with controllable lighting and cameras, it can be very time-consuming, memory-intensive and infeasible in an unconstrained environment. In contrast, our frameworks presented in Chapters 3, 4 and 7 can estimate complex spatially-varying material appearance from a single image in a single forward pass of a deep neural network.

Transparency is an even more challenging material property to estimate, especially with just a few input images – its appearance effects are highly view dependent and complex due to non-local reflection and refraction. As such, although transparent materials are very common in daily life, they are relatively less explored in prior works and largely limited to highly controlled

and dense acquisition setups [133, 106, 214]. In contrast, in Chapter 5, we will present our frameworks for reconstructing transparent shapes from just 5 to 20 views captured in arbitrary indoor environments.

Complex lighting

Lighting estimation is another challenging problem in inverse rendering, primarily due to missing information: (1) Many areas of the scene that contribute to the final radiance may not be directly observed in the image. These areas may include light sources or surfaces that reflect light. (2) Natural lighting has a high dynamic range (HDR) that cannot be directly captured in a low dynamic range (LDR) image, leading to saturated pixels. However, this missing HDR information is crucial for rendering realistic virtual appearances. Besides missing information, heavy occlusions and strong shadows can lead to complex spatially-varying lighting, which is especially true for indoor scenes. The brute-force way to recover spatially-varying HDR lighting is to take multiple images covering every location and view direction, at different exposures to recover the high dynamic range. This is too expensive for large-scale scenes. In contrast, our framework proposed in Chapter 7 can reconstruct spatially-varying HDR indoor lighting from a single LDR image.

What makes lighting estimation even more challenging is the diversity of light sources present in the real world, such as flashlights on cameras, lamps in indoor scenes, sunlight, and sky lighting, which need to be modeled with different representations for both geometry and color spectra. Further, estimating them in images is challenging since their effects often manifest as high-frequency information or distant interactions, even though their spatial extents might be highly localized. In Chapter 3 and 4, we model flashlights on mobile phone cameras as point light sources and utilize them to capture material appearance from a single image. In Chapter 8, we propose different physically-based lighting representations for various indoor light sources.

Complex interactions between scene factors

The interactions between lighting, materials and geometry usually cause non-local light transport effects that make the inverse rendering problem extremely challenging in indoor scenes. For example, the sunlight coming through a window can cause specular highlights on a table and soft shadows on the floor, even though none of the three objects (window, table and floor) might be close to each other in image space or 3D space. These long-range interactions imply that global context information is crucial to accurately recover realistic material and lighting from sparsely sampled images.

Moreover, the light coming from light sources may experience an indefinite number of bounces before it finally reaches the camera. This phenomenon is known as global illumination or interreflection. It can cause color bleeding between surfaces, which further adds difficulty in disentangling lighting, materials and geometry. Although it is often ignored in prior works, global illumination can contribute substantially to total illumination, especially when the scene geometry is mostly concave (generally true for indoor scenes). In Chapters 4, 5 and 8, we propose our physically-motivated learning frameworks that explicitly model multiple bounces of light transport.

Inverse rendering in indoor scenes

From the above analysis, we may postulate that estimating realistic lighting and materials for indoor scenes is much more challenging than estimating them for a single object in nearly every aspect. Indoor scenes usually present spatially-varying diverse materials that span the entire range of gloss from diffuse to specular, diverse lighting such as lamps of different shapes and spectra and sunlight coming through windows, along with highly complex interreflections and shadows due to visible and invisible objects. In Chapters 6, 7 and 8, we focus on a robust and practical framework for indoor scene geometry, lighting and material reconstruction from a single image, which enables a wide variety of previously intractable photorealistic scene editing applications.

1.1.3 A Computer Vision Perspective

A central goal of computer vision is to help computers understand the visual world. This is an intuitive problem for humans. For example, given an image of an indoor scene, it is easy for us to explain its constituent properties. From a physical perspective, we can easily discern its underlying 3D structure, materials, and lighting that determine image appearances, while reasoning about complex light transport effects. We understand that the shadow under a table is because it occludes the light coming from a lamp, or the specular highlights on a vase are due to its specular material reflecting sunlight coming through a window. However, these are extremely hard problems for a computer to solve and have remained so despite the great progress computer vision has achieved over the past fifty years [198].

This thesis advances physical understanding in computer vision by designing practical frameworks to recover the underlying intrinsic factors – namely geometry, materials and lighting – of images captured under unconstrained conditions. Due to its ill-posed nature, to disambiguate these factors, we need priors that model the physical nature of these intrinsic factors and their complex interactions. Many physically-based vision methods have developed numerous such priors,to which we refer as **model-based** methods [123, 12, 13, 30, 34, 32, 31]. While model-based methods have achieved promising success on various scene reconstruction problems, they still have limitations. First, they usually make assumptions to simplify the complexity of image formation, such as homogeneous material [123], diffuse materials [12, 13], low-frequency [168] or directional lighting [30, 34, 32, 31]. While these assumptions can significantly simplify the optimization process, they may fail to recover complex light transport effects commonly seen in the real world, such as high-frequency directional lighting, interreflections and specular highlights. Second, these methods may heavily depend on hand-crafted features, which limit their ability to accurately model the true physics of real-world conditions. On the contrary, our goal is to develop practical solutions that recover photorealistic materials and lighting of a complex scene from images captured in an unconstrained environment, with complex interactions

between intrinsic factors modeled correctly.

The advent of deep learning brings significant progress to various computer vision problems. Specifically, a convolutional neural network (CNN) possesses the properties that make it a natural and powerful tool for accurately recovering materials, lighting and geometry of complex scenes. It provides an effective way to learn high-quality priors from large-scale data that are essential for solving highly ill-posed scene reconstruction problems. However, directly adopting CNNs is non-trivial and can bring about new challenges, whereby complex, non-local light transport effects such as hard and soft shadows, intereflections and color bleeding may prevent even CNNs from accurately decomposing an image into its intrinsic factors. The high dimensional spatially-varying materials and lighting may require high network capacity, making networks memory intensive and difficult to train. Furthermore, large-scale datasets with high-quality ground truth are essential for CNNs to generalize well, but collecting such a dataset for complex scenes with ground-truth materials and lighting is extremely hard or even intractable. As a result, most **deep learning-based** scene reconstruction methods typically focus solely on geometry reconstruction [58, 57] or adopt an over-simplified image formation assumption, such as intrinsic decomposition [114], which does not suffice to model complex light transport or reconstruct photorealistic materials and lighting [63, 66].

1.1.4 A Computer Graphics Perspective

The task of recovering lighting, geometry, and materials and modeling complex light transport from images is known as **inverse rendering** in computer graphics [131], which has remained a central problem in computer graphics for over twenty years. While computer vision has the goal of reconstructing the constituent factors of image formation, the purpose of inverse rendering in computer graphics is to aid in synthesizing photorealistic images – since the physics of image synthesis has been well-established, the next challenge in achieving greater photorealism is to accurately capture real-world materials, lighting, and geometry. Therefore, inverse rendering in computer graphics often seeks the highest possible accuracy of real-world

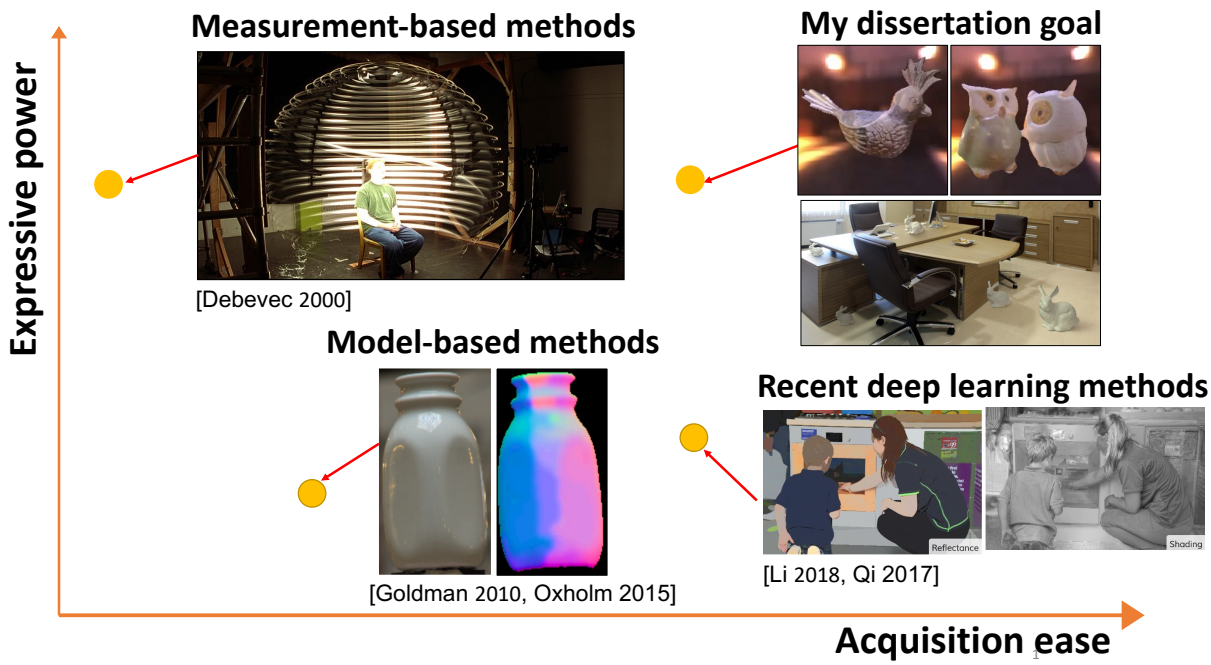


Figure 1.4. A summary of existing inverse rendering frameworks. This thesis motivates a new class of methods that occupy the top-right corner, by combining the advantages of physically-based modeling and advanced deep learning, which achieves high-quality appearance capturing with minimal acquisition difficulties.

intrinsic factors by building large, expensive and carefully calibrated devices, such as light stages and gantries, that can densely sample views and lighting directions [53, 240, 67]. These devices can create controlled environments where two of the three intrinsic factors are already known, so the third component can be recovered accurately. We will refer to these methods as **measurement-based** methods. As these measurement-based methods achieved tremendous success, it is common nowadays to see virtual characters or scenery with realistic, complex virtual appearances in movies and games nearly indistinguishable from the real world.

While the aforementioned inverse rendering methods are effective for creating realistic effects in games and movies, they require expensive devices and controlled environments that are only accessible to experts. This thesis solves an important challenge of computer graphics by designing inverse rendering frameworks that can capture realistic appearances from a single or a few images captured under unconstrained conditions. As a result, non-expert users can use such

methods to recover complex scene appearance by simply taking photographs with their mobile phone cameras.

In summary, inverse rendering remains a difficult problem for computer vision and graphics. Prior methods either need carefully calibrated, constrained setups or utilize simplified models that cannot model the full physics of the image formation process. Recent progress in artificial intelligence, namely deep learning, brings us a new powerful tool to tackle this classical challenge but still faces the issue of generalization ability and lack of training data. In the next section, we will introduce our solution for this canonical challenge, which lays the foundation of this thesis. We propose to build a framework of **physically-motivated learning** for scene reconstruction with complex materials and lighting under unconstrained environments. The key is to instill the inductive bias of the image formation process into the network design, while exploiting domain knowledge to devise meaningful representations and create large-scale photorealistic datasets that make learning tractable. Figure 1.4 summarizes inverse rendering methods based on their acquisition ease and expressive power, where this thesis occupies the top right corner by reasoning about photorealistic complex appearance with extremely sparse inputs, such as a single image captured in the wild.

1.2 Principles of Physically-Motivated Deep Learing

Physically-motivated deep networks

Rather than adopting a purely data-driven approach with black-box CNNs, we propose to create physically-motivated networks for complex scene reconstruction by instilling the domain knowledge of image formation into the network design and training. Such choices allow us to build networks whose outputs are more interpretable, which are better generalizable to real scenes through incorporation of meaningful inductive biases that model the physical nature of intrinsic scene factors and their non-local interactions. Specifically, we introduce in-network, differentiable rendering modules, which allow computing a rendering loss by first synthesizing images from the predicted intrinsic factors and then back-propagating the difference between

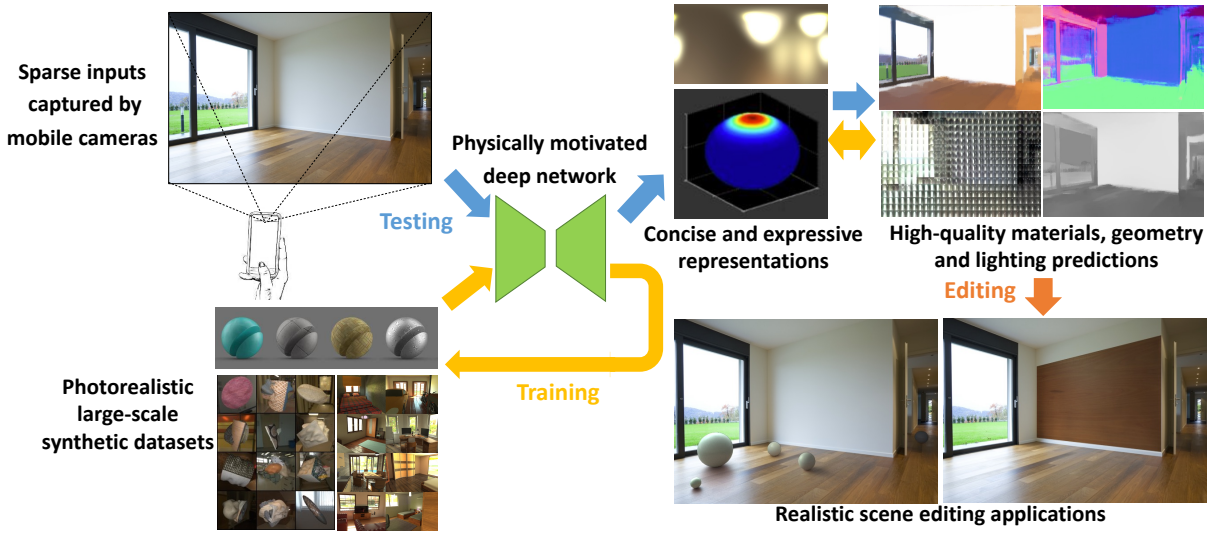


Figure 1.5. A demonstration of our frameworks for solving inverse rendering problems for complex scenes. They key insight is to follow the the physics of the image formation process when creating synthetic datasets, design network architectures and representations so that our framework can achieve high-quality predictions of complex materials and lighting from sparse inputs captured in arbitrary environments. Through editing our predicted intrinsic factors, we may enable photorealistic scene editing applications.

the rendered and input images. These neural rendering modules provide additional supervision to balance the contributions of different intrinsic factors, enabling us to directly optimize for photorealistic final appearance. More importantly, they enable networks to accurately model non-local light transport effects, which are extremely challenging for conventional CNNs. Such effects include color-bleeding due to interreflections in concave scenes (Chapters 4 and 8), refraction and reflection for transparent objects (Chapter 5), and strong soft or hard shadows for indoor scenes (Chapter 8).

Concise and effective representations

The high dimensionality of spatially-varying materials and lighting also present challenges for adopting deep learning-based methods – for example, a traditional representation for lighting and materials as a large set of sampled points covering the whole sphere or hemisphere makes training predictive networks highly memory-intensive. To solve this challenge, we bridge classical works in physically-based rendering and appearance modeling with recent advances in

deep learning, by proposing concise representations that can greatly reduce the dimensionality while being expressive enough for creating photorealistic appearance. Such examples include materials for which we use a physically-based parametric BRDF model [95] that only requires 4 parameters to represent realistic highly-specular materials (Chapters 3, 4 and 7), complex spatially-varying indoor lighting which we represent with the sum of a small number of spherical Gaussian lobes and physical light sources for which we estimate precise geometric extents and color spectra (Chapters 8 and Chapter 7).

Photorealistic synthetic datasets

It is well-known that training CNNs successfully requires a large amount of data. However, collecting large-scale data with accurate ground-truth for spatially-varying materials and lighting for complex scenes can be prohibitively difficult, if not impossible for complex effects such as interreflections and shadows. Thereby, we propose to train on synthetic data where ground-truth materials and light transport are easily available. However, one notorious challenge of training CNNs on synthetic data is that the trained networks may not generalize well to real data due to the differences in data distributions, known as domain gap. Such domain gaps have been addressed in transfer learning with domain adaptation methods, which align the distributions between real and synthetic data. While such approaches remain interesting future directions to pursue, we propose to alleviate domain gaps through large-scale photorealistic synthetic datasets that accurately model the physics of the image formation process. In this way, the priors learned by CNNs on synthetic data are physically meaningful and therefore can generalize well to real scenes. Our twofold contributions have wide potential impacts. Firstly, our open-source tools and data allow even non-expert users to create large-scale datasets for inverse rendering problems using their own images and scans. Secondly, CNNs trained on such photorealistic synthetic datasets achieve high-quality scene reconstruction on real data to enable a wide variety of photorealistic scene editing applications from even a single mobile phone image.

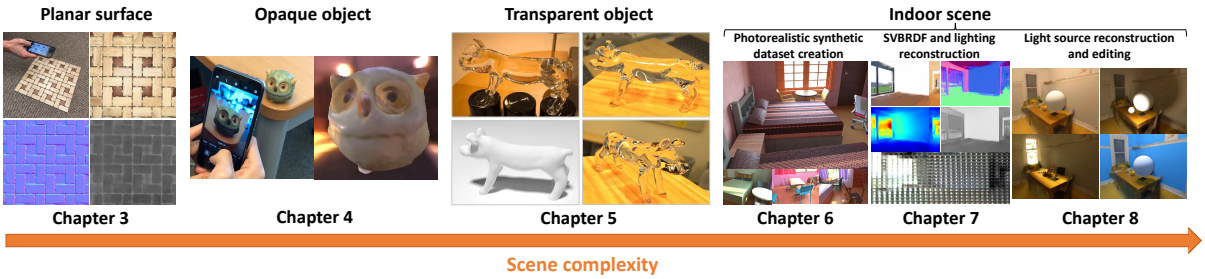


Figure 1.6. Organization of this thesis. We start from relatively simpler setting, helping readers to get familiar with import concepts, to the most difficult setting for inverse rendering.

1.3 Summary

In summary, this thesis develops physically-motivated deep learning frameworks for scene reconstruction with complex lighting and materials, using only sparse inputs captured in an unconstrained environment. The key is to incorporate the inductive bias of image formation when designing physically-motivated networks, devising concise and expressive representations for intrinsic scene components and complex light transport, and creating photorealistic synthetic datasets that utilize such domain knowledge to allow generalization to real scenes. Our approach is summarized in Figure 1.5. This explicit consideration of the physical basis of image formation allows us to achieve high-quality scene reconstruction of geometry, spatially-varying materials and complex lighting that cannot be adequately handled by prior measurement-based, model-based, or learning-based methods. By editing our high-quality predicted intrinsic scene components, we achieve photorealistic effects such as virtual object insertion, light source editing, view synthesis and relighting, which may lay the foundation for the next generation of augmented reality applications.

1.4 Organization of This Thesis

The organization of this thesis is as follows. In Chapter 2, we first introduce background knowledge about image formation and its constituent factors, as well as the table of notation, which is useful to understand this thesis. From Chapter 3 to 8, we introduce our physically-

motivated deep learning frameworks for various inverse rendering problems, starting from the relatively simpler setting of a planar surface, to the most difficult setting of an arbitrary indoor scene.

In Chapter 3, we propose our framework for recovering spatially-varying material appearance of a planar surface from a single image. Even though the geometry is known, the high-dimensional spatially-varying material, which we model as a bidirectional reflectance distribution function (BRDF), is still very difficult to reconstruct. We follow the three above principles to design our physically-motivated deep learning framework and achieve material reconstruction accuracy better than prior state-of-the-art.

In Chapter 4, we move further to jointly estimate geometry and spatially-varying material for a single object from a single image. Compared to planar surfaces, the concave geometry of an arbitrary object can cause non-local interreflections, adding difficulty to disambiguate geometry, materials and lighting. We propose a customized in-network rendering layer that can handle global illumination efficiently and a cascade structure that can iteratively refine our predictions utilizing global context information.

Transparent materials induce complex refraction and reflection, which can be considered as an even more challenging scenario. In Chapter 5, we propose our physically-motivated framework for transparent shape reconstruction from as few as 5 to 20 images captured in arbitrary indoor environments. Two key advances in our framework are: (1) a novel representation that enables an in-network differentiable rendering layer to model the full physics of two-bounce light transport and (2) a cost-volume-based physically-motivated network architecture that can regularize the shape reconstruction.

As discussed in Section 1.1.2, indoor scene appearance reconstruction presents one of the steepest challenges in nearly every aspect. Chapter 6, 7 and 8 propose a practical framework for this extremely difficult problem. We start from Chapter 6 by designing an open framework for creating large-scale photorealistic synthetic indoor datasets. The created dataset is used to train our physically-motivated networks in Chapter 7 and 8.

In Chapter 7, we propose a framework that jointly estimates geometry, spatially-varying material and spatially-varying lighting of an indoor scene from a single image. Directly predicting lighting of every incoming direction at every location can be prohibitively expensive. We adopt a spatially-varying spherical Gaussian (SVSG) that allows us to recover high-frequency directional lighting with a much smaller number of parameters, with global illumination also being modeled correctly. Our frameworks achieve high-quality inverse rendering results that enable various photorealistic scene editing applications that cannot be as well achieved by prior methods, such as virtual object insertion with realistic specular highlights and shadows.

Finally, we achieve the first result of its kind on editing of light sources in complex indoor scenes with accurate non-local light transport – to create effects such as opening a virtual window to let the sunlight in, putting a virtual lamp into the scene, or changing the wall color with consistent color bleeding on all other furniture. This is presented in Chapter 8, where we propose a reconstruction framework that can estimate physically-based visible and invisible light sources in the scene and a neural rendering framework that can re-render the images from our predictions with various complex light transport effects being handled explicitly, such as shadows and global illumination, even with occluded and invisible scene surfaces.

We conclude in Section 9 by revisiting the philosophies espoused by this dissertation, as well as highlighting the several directions of future research opened up by our study of physically-motivated learning for photorealistic scene reconstruction and editing in the wild.

Chapter 2

Background

In this chapter, we will briefly recap background knowledge that will be useful to understand this thesis. We first present the formal definition of image formation by introducing the rendering equation. This brings in the mathematical explanation of several complex light transport effects that will be handled in later chapters. Then, we introduce properties of each intrinsic factor. For materials, we discuss several physically-based parameterized material appearance models that can represent appearance with only a small number of parameters. For lighting, we first discuss several commonly used light sources, including physically-based light sources and image-based lighting, and then introduce two lighting representations for reducing high dimensional lighting. For geometry, we briefly talk about the geometry representation used in this thesis while referring some very recent progress as future work. Finally, we write the table of notation that will be used through the rest of the thesis.

2.1 Image Formation

The complex image formation process is probably best described mathematically by the rendering equation, which was first proposed by Kajiya in his seminal work [93]. Let $\mathbf{p}_i$ be the surface point and $\mathbf{p}_j \rightarrow \mathbf{p}_i$ be the unit vector pointing from $\mathbf{p}_j$ to $\mathbf{p}_i$. Then, the area integral form

of the rendering equation can be written as

$$\mathbf{L}(\mathbf{p}_1 \rightarrow \mathbf{p}_0) = \mathbf{L}_e(\mathbf{p}_1 \rightarrow \mathbf{p}_0) + \int_{\mathscr{A}} \rho(\mathbf{p}_1 \rightarrow \mathbf{p}_0, \mathbf{p}_1 \rightarrow \mathbf{p}_2) G(\mathbf{p}_1, \mathbf{p}_2) \mathbf{L}(\mathbf{p}_2 \rightarrow \mathbf{p}_1) \mathrm{d}\mathscr{A}(\mathbf{p}_2), \quad (2.1)$$

where $\mathbf{L}(\mathbf{p}_i \rightarrow \mathbf{p}_j)$ represents the total radiance coming from $\mathbf{p}_i$ to $\mathbf{p}_j$, including reflection and emission, and $\mathbf{L}_e(\mathbf{p}_i \rightarrow \mathbf{p}_j)$ represents radiance directly emitted by light sources. If surface point $\mathbf{p}_i$ is not on an emitter, then $\mathbf{L}_e(\mathbf{p}_i \rightarrow \mathbf{p}_j)$ should be 0. ρ is the function that decides how materials reflect incoming radiance, namely the material appearance. $\mathscr{A}$ represents all the surface area of the scene. Finally, $G(\mathbf{p}_i, \mathbf{p}_j)$ is the geometric function that models the foreshortening and shadow effects.

$$G(\mathbf{p}_i, \mathbf{p}_j) = \frac{V(\mathbf{p}_i, \mathbf{p}_j) \max\left(\mathbf{p}_i \rightarrow \mathbf{p}_j \cdot \mathbf{N}(\mathbf{p}_i), 0\right) \max\left(\mathbf{p}_j \rightarrow \mathbf{p}_i \cdot \mathbf{N}(\mathbf{p}_j), 0\right)}{||\mathbf{p}_i - \mathbf{p}_j||_2^2} \quad (2.2)$$

where $\mathbf{N}(\cdot)$ is the normal direction and $V(\cdot, \cdot)$ is the binary visibility function telling if a ray is occluded. In the following, we explain several commonly seen complex light transport effects based on the rendering equation (2.1).

Foreshortening

The foreshortening effect is modeled by $\left(\mathbf{p}_i \rightarrow \mathbf{p}_j \cdot \mathbf{N}(\mathbf{p}_i)\right)\left(\mathbf{p}_j \rightarrow \mathbf{p}_i \cdot \mathbf{N}(\mathbf{p}_j)\right)$ in the geometry function $G(\cdot, \cdot)$ (2.2). Intuitively, if the surface normal $\mathbf{N}$ orients towards perpendicular to the view direction, the surface area will look smaller and its contribution to the final radiance decrease. Figure 2.1 shows an example of foreshortening effect (first column). As we rotate the area light source perpendicular to the surface below, the surface center becomes dark. This effect is used by photometric stereo methods [198] to reconstruct the surface normal. In our frameworks in Chapter 3, 4 and 7, we disambiguate the foreshortening effect from lighting and material appearance by jointly reasoning lighting, geometry and material of scenes through training physically-motivated deep networks.

Shadows

Shadows are caused by the occlusion of light sources, which is modeled by the visibility function $V(\cdot,\cdot)$ in (2.2). Most shadows in the real world are soft shadows, i.e., shadows with soft boundaries. This is because the rendering equation is an integral of all incoming radiance – light sources with area may only be partially occluded by scene geometry – causing the shadow boundaries to be soft. Figure 2.1 (second column) demonstrates the shadow effects, which shows that the larger the area of light sources, the softer the shadow boundaries.

Modeling shadows, especially soft shadows, is very challenging because it requires accurate reconstruction of both light sources and occluders gometry and modeling of non-local light transport. In Chapter 8, we solve this challenge by combining the advantages of deep learning on hallucinating unseen geometry and physically-based rendering on accurately modeling long-range occlusions.

Long range interactions

The integral in the rendering equation (2.1) involves the whole scene, which suggests that surfaces not adjacent with each other may contribute significantly to the incoming radiance. Modeling such complex long-range interactions may be too challenging even for recent advanced CNN architectures. In this thesis, we propose in-network differentiable rendering modules to help soling this challenge. In Chapter 3, 4, 7 and 8, such rendering modules are used to compute rendering losses as additional supervision, which help instill networks with domain knowledge of the image formation process.

Global illumination

The rendering equation is actually a recursive function, where $L(\mathbf{p}_2 \rightarrow \mathbf{p}_1)$ in (2.1) can be further decomposed into emission and reflection.

$$\begin{aligned}\mathbf{L}(\mathbf{p}_1 \rightarrow \mathbf{p}_0) &= \mathbf{L}_e(\mathbf{p}_1 \rightarrow \mathbf{p}_0) + \int_{\mathscr{A}} \rho(\mathbf{p}_1 \rightarrow \mathbf{p}_0, \mathbf{p}_1 \rightarrow \mathbf{p}_2) G(\mathbf{p}_1, \mathbf{p}_2) \mathbf{L}_e(\mathbf{p}_2 \rightarrow \mathbf{p}_1) \mathrm{d}\mathscr{A}(\mathbf{p}_2) \\ &+ \int_{\mathscr{A}} \int_{\mathscr{A}} \rho(\mathbf{p}_1 \rightarrow \mathbf{p}_0, \mathbf{p}_1 \rightarrow \mathbf{p}_2) G(\mathbf{p}_1, \mathbf{p}_2) \\ &\quad \rho(\mathbf{p}_2 \rightarrow \mathbf{p}_1, \mathbf{p}_2 \rightarrow \mathbf{p}_3) G(\mathbf{p}_2, \mathbf{p}_3) \mathbf{L}_e(\mathbf{p}_3 \rightarrow \mathbf{p}_2) \mathrm{d}\mathscr{A}(\mathbf{p}2) \mathrm{d}\mathscr{A}(\mathbf{p}3) + ...\end{aligned}$$

The recursive form of the rendering equation describes how light coming from light sources can experience multiple reflections before finally reaching the camera. We refer to this part of lighting as global illumination or interreflection. Global illumination can cause non-local color bleeding; once the light is reflected by the surface, its color will be modulated by the surface material, causing the color from one surface to bleed onto another. This is shown in the third column of Figure 2.1, where the green color from the outside box bleeds to the white sphere. Since global illumination can take a substantial part in the total illumination, we need to model it when solving scene reconstruction problems with concave geometry. However, due to the computational difficulties of handling global illumination, most prior methods neglect this phenomenon. In Chapter 4 and 8, we design physically-motivated network modules that can render complex global illumination efficiently and accurately through a single forward pass. In Chapter 7, we bake global illumination into our local lighting representation so that the color bleeding effect can be rendered efficiently through local computation.

2.2 Materials

Material appearance is described by the function $\rho(\cdot, \cdot)$ in the rendering equation (2.1). This function is called bidirectional reflectance distribution function, shortened as BRDF in the rest of this thesis. Given the incoming and outgoing directions as inputs, it describes how

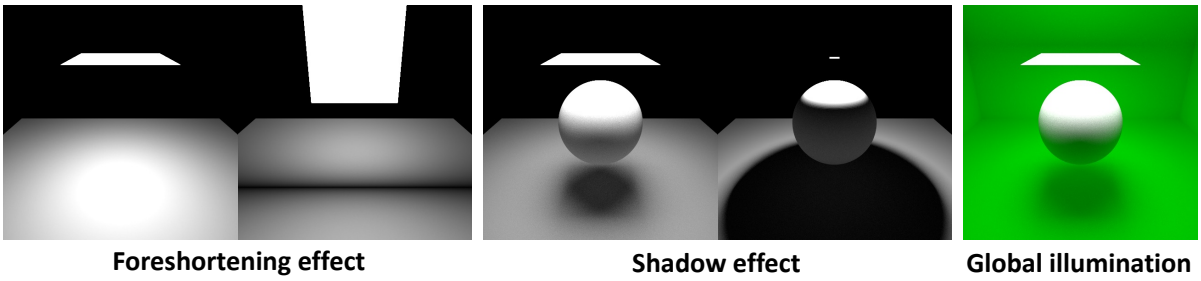


Figure 2.1. A demonstration of some light transport effects, such as foreshortening, soft/hard shadows, and global illumination.

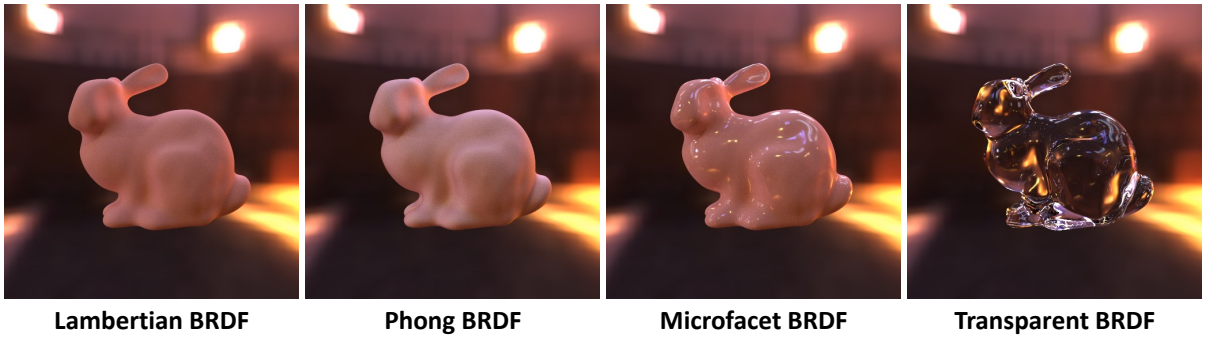


Figure 2.2. A demonstration of some parameterized BRDFs, including Lambertian, phong, microfacet and transparent BRDFs. We observe that microfacet BRDF can create the most realistic specular highlights, while transparent BRDF leads to complex reflection and refraction.

the ray is reflected by materials. The most naive and brute-force way to represent BRDF will be densely sampling every lighting and view direction covering the whole hemisphere [132], which is computationally expensive. Fortunately, the angular distributions of natural real-world BRDFs are not arbitrary. Instead, they usually lie on a low-dimensional manifold that can be well-approximated through physically-based parameterized BRDF models. In the following, we will introduce several popular parameterized BRDF models that allow us to model material appearances with only a small number of parameters.

Lambertian BRDF

The simplest BRDF is a constant number. It means that the appearance of materials is view independent, i.e. it looks the same as we change viewpoints. This BRDF model is known

as Lambertian model and the constant number is material's diffuse albedo ($\mathbf{A}$).

$$\rho(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2; \mathbf{A}) = \frac{\mathbf{A}}{\pi} \tag{2.3}$$

where $\mathbf{A}$ is a 3-channel variable in the range from 0 to 1. π is the normalization factor. Due to its simplicity, Lambertian BRDF is assumed for many classical scene reconstruction tasks, such as multi-view stereo and photometric stereo. However, real-world materials can be highly view dependent, causing complex specular highlight and reflections that cannot be handled by Lambertian BRDF. We will introduce two popular parameteric BRDF models for specular materials.

Phong BRDF

One early attempt to model the view-dependent specularity of materials is the Phong BRDF model [162]. The Phong BRDF model is based on the empirical observation that highly specular materials have sharp and bright specular highlights while less specular materials have blurry and smooth specular highlights. Its specularity is controlled by two variables, the specular color variable ($\mathbf{A}_s$) and the exponential variable s. The formal definition of the Phong BRDF model is

$$\rho(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2; \mathbf{A}, \mathbf{A}_s, s) = \frac{\mathbf{A}}{\pi} + \frac{\mathbf{A}_s(s+2)}{2\pi}(\omega_r \cdot \mathbf{p}_1 \to \mathbf{p}_2)^s \tag{2.4}$$

where ω_r is the reflection direction of $\mathbf{p}_0 \to \mathbf{p}_1$, i.e.

$$\omega_r = 2(\mathbf{p}_1 \to \mathbf{p}_0 \cdot \mathbf{N})\mathbf{N} - \mathbf{p}_1 \to \mathbf{p}_0. \tag{2.5}$$

Note that $\mathbf{A} + \mathbf{A}_s$ has to be smaller than 1 for energy conservation. Intuitively, when s is large, the value of the specular term will drop steeply as the $\mathbf{p}_1 \to \mathbf{p}_2$ moving away from the reflection direction ω_r, causing sharp and strong specular highlights.

While Phong BRDF model is widely used as a baseline model for specularity modeling,

it is an empirical rather than a physically-based BRDF model. Comprehensive experiments find that its error when fitting the real-world material appearance are consistently higher compared to more advanced models with clear physical motivations [148]. In the following, we will introduce a more advanced model that has been used through out the rest of the thesis.

Microfacet BRDF

The microfacet BRDF model [159] is a physically-based BRDF model that can model specular reflectance realistically. It has Lambertian term and specular term. Similarly, its Lambertian term is just the constant diffuse albedo (**A**). Its specular term is controlled by roughness (**R**), which is a 1-channel variable. Intuitively, when the roughness value is small, the material appearance will be more view-dependent, namely more specular. The specular term of the microfacet BRDF model assumes that specular appearances can be represented by a large collection of micro facets with different orientations. Therefore, material appearance is controlled by three factors of these micro facets: the statistical distribution of facets' orientations, how each facet reflects light, and the foreshortening and shadow effects caused by the interactions between facets. These three factors correspond to three functions in the specular term, the distribution function (D), the Fresnel function (F), and the geometry function (G). Without diving into details, the formal definition of the microfacet BRDF model [95] used in this thesis is

$$\begin{aligned}
\rho(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2; \mathbf{A}, \mathbf{R}) &= \frac{\mathbf{A}}{\pi} + \frac{D(\mathbf{h}, \mathbf{R}) F(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{h}) G(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2, \mathbf{h}, \mathbf{R})}{4(\mathbf{N} \cdot \mathbf{l})(\mathbf{N} \cdot \mathbf{v})} \quad (2.6) \\
D(\mathbf{h}, \mathbf{R}) &= \frac{\mathbf{R}^4}{\pi \left[(\mathbf{N} \cdot \mathbf{h})^2 (\mathbf{R}^4 - 1) + 1 \right]^2} \\
F(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{h}) &= (1 - F_0) 2^{-[5.55473(\mathbf{p}_1 \to \mathbf{p}_0 \cdot \mathbf{h}) - 6.8316](\mathbf{p}_1 \to \mathbf{p}_0 \cdot \mathbf{h})} \\
G(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2, \mathbf{N}, \mathbf{R}) &= \frac{\mathbf{N} \cdot \mathbf{p}_1 \to \mathbf{p}_0}{(\mathbf{N} \cdot \mathbf{p}_1 \to \mathbf{p}_0)(1 - \kappa) + \kappa} \frac{\mathbf{N} \cdot \mathbf{p}_1 \to \mathbf{p}_2}{(\mathbf{N} \cdot \mathbf{p}_1 \to \mathbf{p}_2)(1 - \kappa) + \kappa}
\end{aligned}$$

Here $\kappa = (\mathbf{R} + 1)^2 / 8$. $\mathbf{h}$ is the unit-length half-angle vector in the direction of $(\mathbf{p}_1 \to \mathbf{p}_0 + \mathbf{p}_1 \to \mathbf{p}_2)/2$. For the Fresnel function, we set $F_0 = 0.05$ as suggested in [95]. We use microfacet BRDF for all frameworks in this thesis to reconstruct realistic specularity. Particularly, in Chapter 6,

we build a new large-scale, synthetic indoor dataset by adopting microfacet BRDF for material appearance, which helps us to achieve a new-level of photorealism that can hardly be achieved by prior works. Figure 2.2 compares the appearance of an object rendered with the above three BRDF models, where we observe that microfacet BRDF can model the most realistic specular highlight. More comparisons can be found in Chapter 6.

Transparent BRDF

One special kind of BRDF is that of transparent materials. The BRDF of transparent materials are the sum of two δ functions centered at refraction and reflection directions, i.e. only lighting coming from the two directions can contribute to transparent material appearances. As a result, the rendering equation will degrade into

$$\begin{aligned} \mathbf{L}(\mathbf{p}_1 \rightarrow \mathbf{p}_0) &= F(\omega_r, \omega_t; \eta)\mathbf{L}(\omega_r) + (1 - F(\omega_r, \omega_t; \eta))\mathbf{L}(\omega_t) \\ F(\omega_r, \omega_t; \eta) &= \frac{1}{2}\left(\frac{\omega_r \cdot \mathbf{N} - \eta\,\omega_t \cdot \mathbf{N}}{\omega_i \cdot \mathbf{N} + \eta\,\omega_t \cdot \mathbf{N}}\right)^2 + \frac{1}{2}\left(\frac{\eta\,\omega_i \cdot \mathbf{N} - \omega_t \cdot \mathbf{N}}{\eta\,\omega_i \cdot \mathbf{N} + \omega_t \cdot \mathbf{N}}\right)^2. \end{aligned} \tag{2.7}$$

where ω_r is the reflection direction as computed in Eq. 2.5 and ω_t is the refraction direction computed following Snell's law. η is the index refraction (IoR) of the transparent material.

Shape reconstruction with transparent materials is an extremely challenging problem due to the complex light paths induced by reflection and refraction, as shown in Figure 2.2. In Chapter 5, We propose the first physically-motivated network to recover 3D shape of transparent objects using a few images, captured in unconstrained indoor scenes.

2.3 Lighting

In the following, we introduce four types commonly used light sources. We start from physically-based emitters to image-based lighting. Finally, we introduce two concise and expressive representations that can effectively reduce the dimensionality of lighting.

Point light

Point light is a simplified model which assumes that the light source is infinite small. Therefore, the surface integral of the rendering equation (2.1) will degrade into

$$\mathbf{L}(\mathbf{p}_1 \to \mathbf{p}_0) = \rho(\mathbf{p}_1 \to \mathbf{p}_0, \mathbf{p}_1 \to \mathbf{p}_2) G(\mathbf{p}_1, \mathbf{p}_2) \mathbf{L}_e(\mathbf{p}_2) \tag{2.8}$$

where $\mathbf{p}_2$ is the location of the point light sources and $\mathbf{L}_e(\mathbf{p}_2)$ is its intensity. The geometry function G now degrades into

$$G(\mathbf{p}_1, \mathbf{p}_2) = \frac{V(\mathbf{p}_1, \mathbf{p}_2) \max\left(\mathbf{p}_1 \to \mathbf{p}_2 \cdot \mathbf{N}(\mathbf{p}_1), 0\right)}{||\mathbf{p}_1 - \mathbf{p}_2||_2^2} \tag{2.9}$$

Note that compared to Eq. (2.2), $\mathbf{p}_2 \to \mathbf{p}_1 \cdot \mathbf{N}(\mathbf{p}_2)$ is removed because the point light source has no surface normal. Point light source can be used to model the flashlight on cameras. In Chapter 3 and 4, we turn on the flashlight when capturing spatially-varying BRDF of a planar surface and a single object, to simplify the lighting condition and achieve more accurate BRDF reconstruction. However, point light source is only a good approximation for very small light sources and cannot be used for large light sources with arbitrary geometry. Besides, it cannot create soft shadows, which is very common in natural environments. These issue can be solved by using area light representation.

Area light

Area light model is a common and simple model for light sources with arbitrary geometry. It assumes that light sources emit light uniformly in every direction covering the exterior hemisphere of their arbitrary surfaces. In Chapter 6 and Chapter 8, we use this model to approximate lamps in indoor scenes, with effects such as occlusion and soft shadows being handled properly.

However, area light model cannot handle directional lighting, such as sunlight coming through windows, which is very common for indoor scenes. To handle this, we will propose

Figure 2.3. Environment map representation. From the left to right is a demonstration of the parameterization of environment maps and a mirror sphere rendered with environment lighting.

a simple parameterized sky lighting model in Chapter 8 that explicitly models high-frequency directional sunlight as well as more ambient lighting from sky and ground.

As discussed earlier, light coming from the above physically-based emitters may experience an indefinite number of bounces before reaching the camera. Directly modeling this global illumination effect can be computationally expensive and may require a full reconstruction of the scene's geometry and materials, which is almost intractable when we only have sparse observation of the scene. There are two potential methods to fix this issue. One is to utilize the powerful hallucination ability of deep network to directly predict global illumination from partial reconstruction of the scene, as proposed in Chapter 7 and Chapter 8. The other is that instead of using physically-based emitter, we directly predict the total incoming radiance without separating emission and reflection. We call this lighting representation environment map.

Environment map

Environment map is an HDR image that directly records the total incoming radiance from every direction covering a sphere or a hemisphere. Therefore, it is also called image-based lighting (IBL). Formally, let $\mathbf{E}$ be an environment map image covering the whole sphere, ω_i be the incoming direction and θ, ϕ be ω_i's elevation and azimuth angles in range of $[0, \pi]$ and $[0, 2\pi)$ respectively. The incoming radiance $\mathbf{L}_e$ can be acquired as

$$\mathbf{L}_e(\omega_i) = \mathbf{E}(\frac{\theta}{\pi}, \frac{\phi}{2\pi}) \tag{2.10}$$

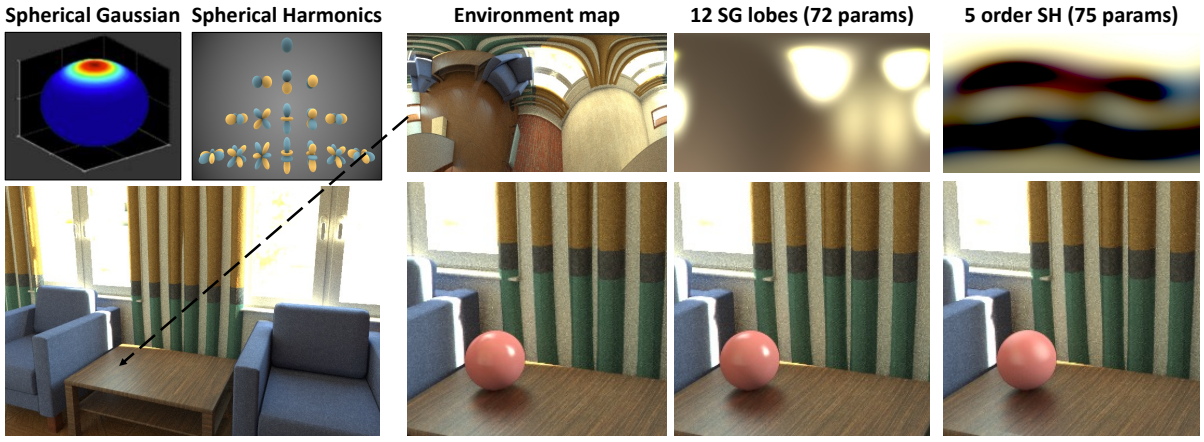


Figure 2.4. Two concise and expressive representations for environment lighting. We observe that with similar number of parameters, Spherical Gaussian representation can better preserve high-frequency lighting, leading to more realistic rendering for specular materials.

To help understanding, this parameterization is also visualized in Figure 2.3. The major advantage of the environment map model is its simplicity. We no longer need to model light sources, visibility and global illumination since they are all baked into the incoming radiance. With the environment map representation, the rendering equation can be simplified as a single integral over the unit sphere, rather than recursive integrals over the whole scene.

$$\mathbf{L}(\mathbf{p}_1 \to \mathbf{p}_0) = \int_{\Omega} \rho(\mathbf{p}_1 \to \mathbf{p}_0, \omega_i) \mathbf{L}_e(\omega_i) \max(\omega_i \cdot \mathbf{N}, 0) \mathrm{d}\omega_i \tag{2.11}$$

However, environment map can be an expensive representation because we may need a large size image to record every ray. Furthermore, environment map does not contain any geometry information. To capture spatially-varying lighting, we may need to sample multiple environment maps at different locations. These above issues motivate us to adopt more concise representations to model incoming radiance, which we will discuss in the following.

Low dimensional lighting representation

The essential reason why we can reduce the dimensionaity of environment lighting can be seen from the rendering equation (2.11). The rendering equation is essentially a convolution

between BRDF and the environment lighting. Therefore, if the BRDF is not extremely high-frequency, such as δ function for transparent materials, some missing details in the environment lighting may not cause significant errors in the final rendered appearance.

Based on this observation, several low dimensional representations have been proposed to approximate environment lighting, which facilitate both lighting estimation and fast rendering of the scene. One of such representations that is widely used in computer vision and graphics is Spherical Harmonics (SH). Spherical Harmonics is the analog of Fourier basis for sphere functions [167]. It allows us to represent an environment map with a smaller number of coefficients of its basis function. Formally, let $\{Y_{nm}(\omega_i)\}$ be the basis functions. Their coefficient $\{k_{nm}\}$ are computed as

$$k_{nm} = \int_{\Omega} Y_{nm}(\omega_i)\mathbf{L}(\omega_i)\mathrm{d}\omega_i \ . \tag{2.12}$$

The environment lighting can then be approximated as

$$\mathbf{L}_e(\omega_i) = \sum_{n=0}^{N}\sum_{m=-n}^{n} k_{nm}Y_{nm}(\omega_i) \tag{2.13}$$

where N is the order of the Spherical Harmonics. With larger number of N, we can capture more high-frequency signals of environment lighting.

Spherical Harmonics is good at recovering low-frequency signals but need a large number of parameters for high-frequency directional lighting. Since our goal is to model realistic specular highlights and shadows, we adopt a relatively less explored representation, Spherical Gaussian, which better models high-frequency signals. In this representation, we approximate environment lighting as the sum of a small number of Spherical Gaussian lobes. Each of them has its own direction, intensity and bandwidth. Compared to Spherical Harmonics, this representation offers us the flexibility to focus on small bright spots in the environment map so that we can better model directional lighting with less parameters. Formally, let $\mathbf{w}$, λ and $\mathbf{d}$ be the weight, bandwidth and direction of a spherical Gaussian lobe respectively. An environment map can be

Table 2.1. Table of notations

$\mathbf{A}$	Diffuse albedo	$\mathbf{N}$	Normal	$\mathbf{R}$	Roughness	$\mathbf{D}$	Depth
$\mathbf{E}$	Environment map	$\mathbf{S}$	Shadow	$\mathbf{H}$	Diffuse shading	$\mathbf{I}$	Image
$\mathbf{L}$	Light intensity	$\mathbf{M}$	Mask	η	Index of refraction	$\mathscr{L}$	Loss function
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Axis	k	Coefficient	ρ	BRDF	$\mathbf{c}$	Center
$\mathtt{W}$	Window	$\mathtt{L}$	Lamp	$\hat{}$	Ground-truth	$\tilde{}$	Intermediate result
$\mathbf{SG}$	Spherical Gaussian	$\mathbf{w}$	Intensity of $\mathbf{SG}$	λ	Bandwidth of $\mathbf{SG}$	$\mathbf{d}$	Direction of $\mathbf{SG}$
$\mathbf{p}, \mathbf{q}$	3D points	ω	Unit vector	θ	Elevation angle	ϕ	Ezimuth angle

approximated as

$$\mathbf{L}_e(\omega_i) = \sum_{n=0}^{N} \mathbf{w}_n \exp(\lambda_n(\omega_i \mathbf{d}_n - 1)) \tag{2.14}$$

With some abuse of notation, N is the number of Spherical Gaussian lobes. Figure 2.4 compares the SH and SG representations, where we observe that with similar number of parameters, SG representation better preserve high-frequency signal and can be used to render sharper and more realistic specular highlights and shadows. In Chapter 7, we use spatially-varying Spherical Gaussian representation (SVSG) to model spaitally-varying lighting for complex indoor scenes and achieve state-of-the-art accuracy. In Chapter 8, we use Spherical Gaussian to approximate outdoor lighting coming through windows.

2.4 Geometry

In this thesis, we use the most commonly used geometry representation in this thesis, including depth map and mesh. For transparent shape reconstruction specifically, we propose a two-normal geometry representation that supports fast rendering of two-bounce refraction and reflection, which we will discuss in more detail in Chapter 5. There is a recent trend of representing scene geometry through implicit representation [137]. However, this is beyond the scope of this paper and will be discussed more in the Chapter 9.

Table 2.1 summarizes the notations used in this thesis.

Chapter 3

Planar Surface SVBRDF Reconstruction from a Single Image

3.1 Introduction

In this chapter, we propose a deep learning-based material estimation approach to recover the spatially-varying BRDF and normal map of a near-planar surface from a single image captured by a handheld mobile phone camera. While the advent of convolutional neural networks (CNNs) has recently led to significant advances in recovering shape using just a single image [57, 45], material estimation has not seen as much progress, which might be attributed to multiple causes. First, material properties can be more complex. Even discounting more complex global illumination effects, materials are represented by a spatially-varying bidirectional reflectance distribution function (SVBRDF), which is an unknown high-dimensional function that depends on exitant and incident lighting directions. Second, while large-scale synthetic and real datasets have been collected for shape estimation [36, 188], there is a lack of similar data for material estimation. Third, pixel observations in a single image contain entangled information from factors such as shape and lighting, besides material, which makes estimation ill-posed.

Conventional BRDF capture setups usually require significant equipment and expense [53, 130]. In contrast, we address this challenge by proposing aphysically-motivated CNN architecture that is specifically designed to account for the physical form of BRDFs and the interaction of light with materials, which leads to a better learning objective. Specifically, We

first encodes the input image into a latent representation, which is decoded into components corresponding to surface normals, diffuse texture, and specular roughness. The key innovation of our CNN is a differentiable rendering layer that recombines the estimated components with a novel lighting direction. This gives us additional supervision from images of the material rendered under arbitrary lighting directions, which helps us balance the contributions of different material parameters and directly optimize for realistic appearance. We also observe that coarse classification of BRDFs into material meta-categories is an easier task, so we additionally include a material classifier to constrain the latent representation. The inferred BRDF parameters from the CNN are quite accurate, but we achieve further improvement using densely-connected conditional random fields (DCRFs) with novel unary and smoothness terms that reflect the physical properties of the underlying BRDF model.

In addition, we also propose to use a dataset of microfacet SVBRDFs that has been designed for perceptual accuracy of materials. This is in contrast to prior datasets that are limited to homogeneous materials, or conflate material properties with other concepts such as object categories.

Our approach – using our novel architecture and SVBRDF dataset – can outperform the state-of-art. We demonstrate that we can further improve these results by leveraging a form of acquisition control that is present on virtually every mobile phone – the camera flash. We turn on the flash of the mobile phone camera during acquisition; our images are thus captured under a combination of unknown environment illumination and the flash. The flash illumination helps further improve our reconstructions. First, it minimizes shadows caused by occlusions. Second, it allows better observation of high-frequency specular highlights, which allows better characterization of material type and more accurate estimation. Third, it provides a relatively simple setup for acquisition that eases the burden on estimation and allows the use of better post-processing techniques.

In contrast to recent works such as [5] and [6] that can reconstruct BRDFs with stochastic textures, we can handle a much larger class of materials. Also, our results, both with and without

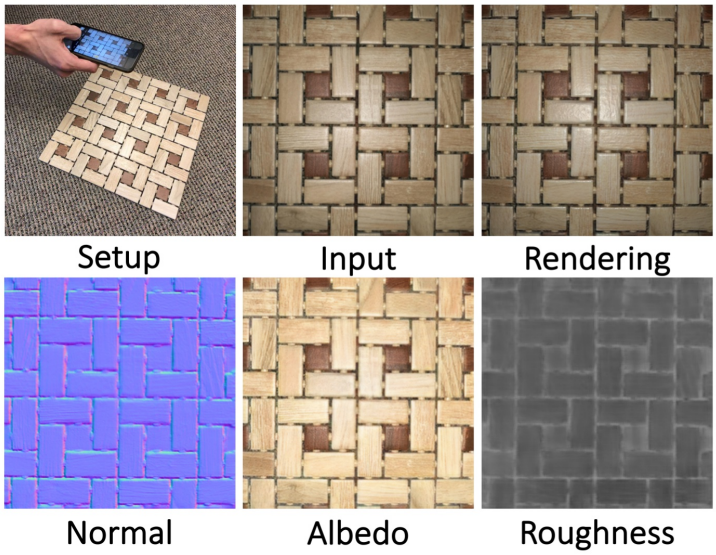


Figure 3.1. We propose a deep learning-based light-weight SVBRDF acquisition system. From a single image of a near planar surface captured with a flash-enabled mobile phone camera under arbitrary lighting, our network recovers surface normals and spatially-varying BRDF parameters – diffuse albedo and specular roughness. Rendering the estimated parameters produces an image almost identical to the input image.

flash, are a significant improvement over the recent method of Li et al. [112] even though our trained model is more compact. Our experiments demonstrate advantages over several baselines and prior works in quantitative comparisons, while also achieving superior qualitative results. In particular, the generalization ability of our network trained on the synthetic BRDF dataset is demonstrated by strong performance on real images, acquired in the wild, in both indoor and outdoor environments, using multiple different phone cameras. Given the estimated BRDF parameters, we also demonstrate applications such as material editing and relighting, which is also shown in the accompanying video [4]. To summarize, we propose the following novel contributions:

- A lightweight method for high quality acquisition of SVBRDF and normal map using a single mobile phone image in an unconstrained environment.
- A physically-motivated CNN and DCRF framework for joint SVBRDF reconstruction and material classification.

- Use of a large-scale SVBRDF dataset specifically attuned to complex materials.

3.2 Related Work

BRDF Acquisition

The Bidirectional Reflection Distribution function (BRDF) is a 4-D function that characterizes how a surface reflects lighting from an incident direction toward an outgoing direction. Alternatively, BRDFs are represented using low-dimensional parametric models [26, 49, 209, 152]. In this work, we use a physically-based microfacet model [96] that our SVBRDF dataset uses.

Traditional methods for BRDF acquisition rely on densely sampling this 4-D space using expensive, calibrated acquisition systems [53, 130, 132]. Recent work has demonstrated that assuming BRDFs lie in a low-dimensional subspace allows for them to be reconstructed from a small set of measurements [149, 225]. However, these measurements still to be taken under controlled settings. We assume a single image captured under largely uncontrolled settings.

Photometric stereo-based methods recover shape and BRDF from images. Some of these methods recover a homogeneous BRDF given one or both of the shape and illumination [174, 175, 155]. Chandraker et al. [32, 31, 33] utilize motion cues to jointly recover shape and BRDF from images under known directional illumination. Hui et al. [81] recover SVBRDFs and shape from multiple images under known illuminations. All of those methods require some form of controlled acquisition, while we estimate SVBRDFs and normal maps "in-the-wild".

Recent work has shown promising results for "in-the-wild" BRDF acquisition. Hui et al. [82] demonstrate that the collocated camera-light setup on mobile devices is sufficient to reconstruct SVBRDFs and normals. They require over 30 calibrated images, while we aim to do the same with a single image. Aittala et al. [6] propose using a flash and no-flash image pair to reconstruct *stochastic* SVBRDFs and normals using an optimization-based scheme. Our method can handle a larger class of materials and is orders of magnitude faster.

Deep learning-based Material Estimation

Inspired by the success of deep learning for a variety of vision and graphics tasks, recent work has considered CNN-based material recognition and estimation. Bell et al. [19] train a material parsing network using crowd-sourced labeled data. However, their material recongition is driven more by object context, rather than appearance. Liu et al. [121] demonstrate image-based material editing using a network trained to recover homogenous BRDFs. Methods have been proposed to decompose images into their intrinsic image components which are an intermediate representation for material and shape [143, 184, 186]. Rematas et al. [170] train a CNN to reconstruct the reflectance map – a convolution of the BRDF with the illumination – from a single image of a shape from a known class. In subsequent work, they disentangle the reflectance map into the BRDF and illumination [65]. Neither of these methods handle SVBRDFs, nor do they recover fine surface normal details. Kim et al. [102] reconstruct a homegeneous BRDF by training a network to aggregate multi-view observations of an object of known shape.

Similar to us, Aittala et al. [5] and Li et al. [112] reconstruct SVBRDFs and surface normals from a single image of a near-planar surface. Aittala et al. use a neural style transfer-based optimization approach to iteratively estimate BRDF parameters, however, they can only handle stationary textures and there is no correspondence between the input image and the reconstructed BRDF [5]. Li et al. use supervised learning to train a CNN to predict SVBRDF and normals from a single image captured under environment illumination [112]. Their training set is small, which necessitates a self-augmentation method to generate training samples from unlabeled real data. Further, they train a different set of networks for each parameter (diffuse texture, normals, specular albedo and roughness) and each material type (wood, metal, plastic). We demonstrate that by using our novel CNN architecture, supervised training on a high-quality dataset and acquisition under flash illumination, we are able to (a) reconstruct all these parameters with a single network, (b) learn a latent representation that also enables material recognition and editing, (c) obtain results that are significantly better qualitatively and quantitatively.

3.3 Acquisition Setup and SVBRDF Dataset

In this section, we describe the setup for single image SVBRDF acquisition and the dataset we use for learning.

Setup

Our goal is to reconstruct the spatially-varying BRDF of a near planar surface from a single image captured by a mobile phone with the flash turned on for illumination. We assume that the z-axis of the camera is approximately perpendicular to the planar surface (we explicitly evaluate against this assumption in our experiments). For most mobile devices, the position of the flash light is usually very close to the position of the camera, which provides us a univariate sampling of a isotropic BRDF [82]. We argue that by imaging with a collocated camera and point light, we can have additional constraints that yield better BRDF reconstructions compared to acquisition under just environment illumination.

Our surface appearance is represented by a microfacet parametric BRDF model [95], whose definition was given in (2.6). It has three parameters, diffuse albedo (**A**), roughness (**R**) and detailed normal map (**N**). Given an observed image **I**, captured under unknown illumination **L**, we wish to recover the parameters **A**, **N** and **R** for each pixel in the image. Please refer to the Chapter 2 for more details on the BRDF model.

Dataset

We train our network on the Adobe Stock 3D Material dataset[1], which contains 688 materials with high resolution (4096×4096) spatially-varying BRDFs. Part of the dataset is created by artists while others are captured using a scanner. We use 588 materials for training and 100 materials for testing. For data augmentation, we randomly crop 12, 8, 4, 2, 1 image patches of size 512, 1024, 2048, 3072, 4096. We resize the image patches to a size of 256×256 for processing by our network. We flip patches along x and y axes and rotate them in increments

[1] https://stock.adobe.com/3d-assets

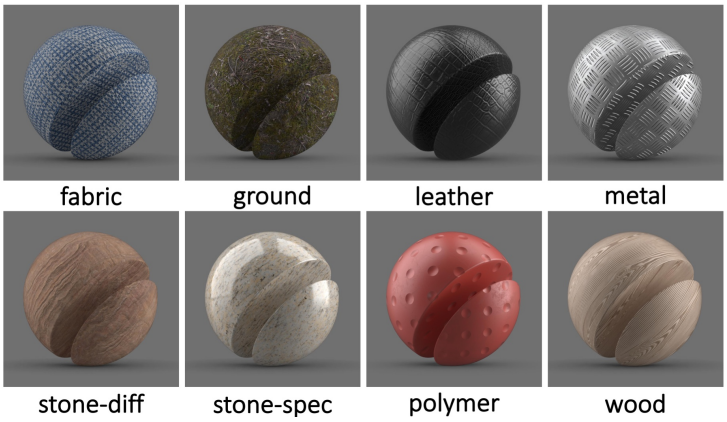


Figure 3.2. Examples of our material types. We divide materials into 8 categories.

Table 3.1. Distribution of materials in our training and test sets.

Materials	Train	Test	Materials	Train	Test
fabric	165	29	polymer	33	6
ground	23	4	stone-diff	177	30
leather	10	2	stone-spec	38	6
metal	82	13	wood	60	10

of 45 degrees. Thus, for each material type, we have 270 image patches.[2] We randomly scale the diffuse color, normal and roughness for each image patch to prevent the network from overfitting and memorizing the materials. We manually segment the dataset into 8 materials types. The distribution is in Table 3.1, with an example visualization of each material type in Figure 3.2.

3.4 Network Design for SVBRDF Estimation

In this section, we describe the components of our CNN designed for single-image SVBRDF estimation. The overall architecture is illustrated in Figure 3.3.

3.4.1 Considerations for Network Architecture

Single-image SVBRDF estimation is an ill-posed problem. Thus, we adopt a data-driven approach with a custom-designed CNN that reflects physical intuitions.

Our basic network architecture consists of a single encoder and three decoders which

[2]The total number of image patches for each material can be computed as $(12+8+4+2+1)\times(1+2+7)=270$.

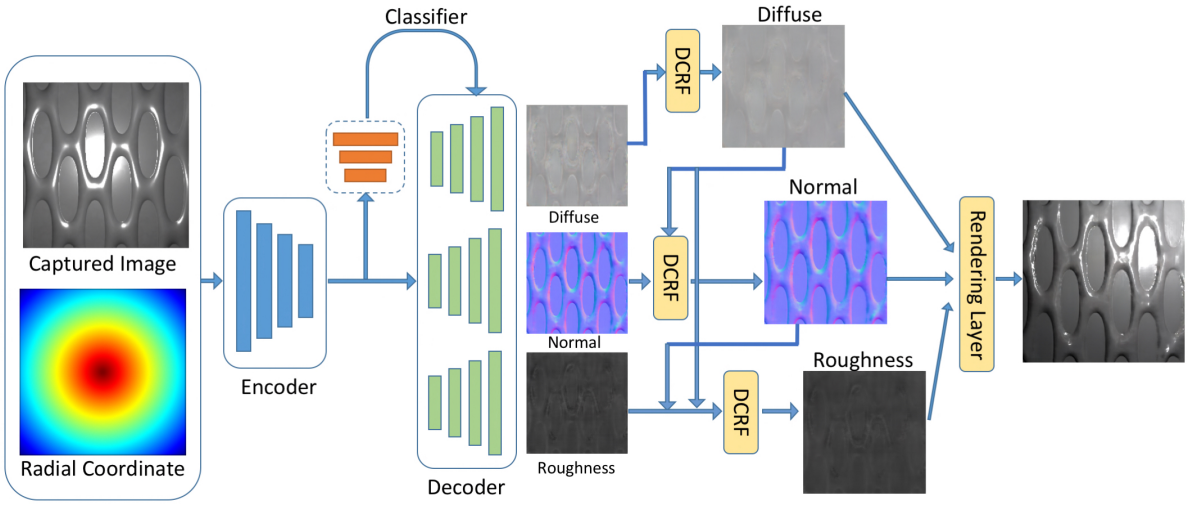


Figure 3.3. Our network for SVBRDF estimation consists of an encoder, three decoder blocks with skip links to retrieve SVBRDF components, a rendering layer and a material classifier, followed by a DCRF for refinement (not visualized). See Section 3.4 for how our architectural choices are influenced by the problem structure of SVBRDF estimation.

reconstruct the three spatially-varying BRDF parameters: diffuse color **A**, normals **N** and roughness **R**. The intuition behind using a single encoder is that different BRDF parameters are correlated, thus, representations learned for one should be useful to infer the others, which allows significant reduction in the size of the network. The input to the network is an RGB image, augmented with the pixel coordinates as a fourth channel. We add the pixel coordinates since the distribution of light intensities is closely related to the location of pixels, for instance, the center of the image will usually be much brighter. Since CNNs are spatially invariant, we need the extra signal to let the network learn to behave differently for pixels at different locations. Skip links are added to connect the encoder and decoders to preserve details of BRDF parameters.

Another important consideration is that in order to model global effects over whole images like light intensity fall-off or large areas of specular highlights, it is necessary for the network to have a large receptive field. To this end, our encoder network has seven convolutional layers of stride 2, so that the receptive field of every output pixel covers the entire image.

3.4.2 Loss Functions for SVBRDF Estimation

For each BRDF parameter, we have an L_2 loss for direct supervision. We now describe other losses for learning a good representation for SVBRDF estimation.

Rendering layer

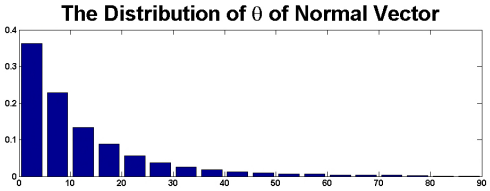
Since our eventual goal is to model the surface appearance, it is important to balance the contributions of different BRDF parameters. Therefore, we introduce a differentiable rendering layer that renders our BRDF model (Eqn. 2.6) under the known input lighting. We add a reconstruction loss based on the difference between these renderings with the predicted parameters and renderings with ground-truth BRDF parameters. The gradient can be backpropagated through the rendering layer to train the network. In addition to rendering the image under the input lighting, we also render images under *novel* lights. For each batch, we create novel lights by randomly sampling the the point light source on the upper hemisphere. This ensures that the network does not overfit to collocated illumination and is able to reproduce appearance under other light conditions. The final loss function for the encoder-decoder part of our network is:

$$\mathscr{L} = k_a\mathscr{L}_a + k_n\mathscr{L}_n + k_r\mathscr{L}_r + k_{rec}\mathscr{L}_{rec}, \tag{3.1}$$

where $\mathscr{L}_a$, $\mathscr{L}_n$, $\mathscr{L}_r$ and $\mathscr{L}_{rec}$ are the L_2 losses for diffuse, normal, roughness and rendered image predictions, respectively. Here, k's are positive coefficients to balance the contributions of various terms, which are set to 1 in our experiments.

Since we train on near planar surfaces, the majority of the normal directions are flat. Table 3.2 shows the normal distributions in our dataset. To prevent the network from over-smoothing the normals, we group the normal directions into different bins and for each bin we assign a different weight when computing the L_2 error. This balance various normal directions in the loss function.

Table 3.2. The θ distribution of the normal vector in the dataset, where θ is the angle between normal vector and $\mathbf{z}$ axis. To avoid the network from over-smoothing the normal map, we group normal vectors into three bins according to θ. With probability P_i for bin i, its weight is $W_i = 0.7 + 1/10P_i$.



Angle	$0°-10°$	$10°-25°$	$25°-90°$
Prob(P_i)	0.592	0.278	0.130
Weight(W_i)	0.869	1.060	1.469

Material Classification

The distribution of BRDF parameters is closely related to the surface material type. However, training separate networks for different material types similar to [112] is expensive. Also the size of the network grows linearly with the number of material types, which limits utility. Instead, we propose a split-merge network with very little computational overhead.

Given the highest level of features extracted by the encoder, we send the feature to a classifier to predict its material type. Then we evaluate the BRDF parameters for each material type and use the classification results as (the output of softmax layer) weights. This averages the prediction from different material types to obtain the final BRDF reconstruction results. Suppose we have N channels for BRDF parameters and J material types. To output the BRDF reconstruction for each type of material, we only modify the last convolutional layer of the decoder so that the output channel will be $J \times N$ instead of N. In practice, we set J to be 8, as shown in Table 3.1.

The classifier is trained together with the encoder and decoder from scratch, with the weights of each label set to be inversely proportional to the number of examples in Table 3.1 to balance different material types in the loss function. The overall loss function of our network with the classifier is

$$\mathscr{L} = k_a\mathscr{L}_a + k_n\mathscr{L}_n + k_r\mathscr{L}_r + k_{rec}\mathscr{L}_{rec} + k_{cls}\mathscr{L}_{cls}, \tag{3.2}$$

where $\mathscr{L}_{cls}$ is cross entropy loss and $k_{cls} = 0.0005$ to limit the gradient magnitude.

3.4.3 Designing DCRFs for Refinement

The prediction of our base network is quite reasonable. However, accuracy may further be enhanced by post-processing through a DCRF (trained end-to-end).

Diffuse color refinement

For diffuse prediction, when capturing the image of specular materials, parts of the surface might be saturated by specular highlight. This can sometimes lead to artifacts in the diffuse color prediction since the network has to hallucinate the diffuse color from nearby pixels. To remove such artifacts, we incorporate a densely connected continuous conditional random field (DCRF) [171] to smooth the diffuse color prediction. Let $\mathbf{A}_i$ be the diffuse color prediction of network at pixel i, $\mathbf{p}_i$ be its position and $\bar{\mathbf{I}}_i$ is the normalized diffuse RGB color of the input image. We use the normalized color of the input image to remove the influence of light intensity when measuring the similarity between two pixels. The energy function of the dense connected CRF that is minimized over $\{\mathbf{A}_i^*\}$ for diffuse prediction is defined as:

$$\sum_{i=1}^{N} \alpha_i^a (\mathbf{A}_i^* - \mathbf{A}_i)^2 + \sum_{i,j}^{N} (\mathbf{A}_i^* - \mathbf{A}_j^*)^2 \big(\beta_1^a \kappa_1(\mathbf{p}_i;\mathbf{p}_j) + \beta_2^a \kappa_2(\mathbf{p}_i,\bar{\mathbf{I}}_i;\mathbf{p}_j,\bar{\mathbf{I}}_j) + \beta_3^a \kappa_3(\mathbf{p}_i,\mathbf{A}_i;\mathbf{p}_j,\mathbf{A}_j)\big)\,.$$

Here κ_i are Gaussian smoothing kernels, while α_i^a and $\{\beta_i^a\}$ are coefficients to balance the contribution of unary and smoothness terms. Notice that we have a spatially varying α_i^a to allow different unary weights for different pixels. The intuition is that artifacts usually occur near the center of images with specular highlights. For those pixels, we should have lower unary weights so that the CRF learns to predict their diffuse color from nearby pixels.

Normal refinement

Once we have the refined diffuse color, we can use it to improve the prediction of other BRDF parameters. To reduce the noise in normal prediction, we use a DCRF with two

smoothness kernels. One is based on the pixel position while the other is a bilateral kernel based on the position of the pixel and the gradient of the diffuse color. The intuition is that pixels with similar diffuse color gradients often have similar normal directions. Let $\mathbf{N}_i$ be the normal predicted by the network. The energy function for normal prediction is defined as

$$\min_{\{\mathbf{n}_i\}} : \sum_{i=1}^{N} \alpha^n (\mathbf{N}_i^* - \mathbf{N}_i)^2 + \sum_{i,j}^{N} (\mathbf{N}_i^* - \mathbf{N}_j^*)^2 \left(\beta_1^n \kappa_1(\mathbf{p}_i; \mathbf{p}_j) + \beta_2^n \kappa_2(\mathbf{p}_i, \Delta\mathbf{A}_i^*; \mathbf{p}_j, \Delta\mathbf{A}_j^*)\right)$$

Roughness refinement

Since we use a collocated light source to illuminate the material, once we have the normal and diffuse color predictions, we can use them to estimate the roughness term by either grid search or using a gradient-based method. However, since the microfacet BRDF model is not convex nor monotonic with respect to the roughness term, there is no guarantee that we can find a global minimum. Also, due to noise from the normal and diffuse predictions, as well as environment lighting, it is difficult to get an accurate roughness prediction using optimization alone, especially when the glossiness in the image is not apparent. Therefore, we propose to combine the output of the network and the optimization method to get a more accurate roughness prediction. We use a DCRF with two unary terms, $\mathbf{R}_i$ and $\dot{\mathbf{R}}_i$, given by the network prediction and the coarse-to-fine grid search method of [82], respectively:

$$\min_{\{\mathbf{R}_i^*\}} : \sum_{i=1}^{N} \alpha_{i0}^r (\mathbf{R}_i^* - \dot{\mathbf{R}}_i)^2 + \alpha_{i1}^r (\mathbf{R}_i^* - \mathbf{R}_i)^2 + \sum_{i,j}^{N} (\mathbf{R}_i^* - \mathbf{R}_j^*)^2 \left(\beta_0 \kappa_0(\mathbf{p}_i; \mathbf{p}_j) + \beta_1 \kappa_1(\mathbf{p}_i, \mathbf{A}_i^*; \mathbf{p}_j, \mathbf{A}_j^*)\right)$$

All DCRF coefficients are learned in an end-to-end manner using [171]. Here, we have a different set of DCRF parameters for each material type to increase model capacity. During both training and testing, the classifier output is used to average the parameters from different material types, to determine the DCRF parameters.

3.5 Experiments

In this section, we demonstrate our method and compare it to baselines on a wide range of synthetic and real data.

Rendering synthetic training dataset

To create our synthetic data, we apply the SVBRDFs on planar surfaces and render them using a GPU based renderer with the BRDF importance sampling suggested in [95]. We choose a camera field of view of 43.35° to mimic typical mobile phone cameras. To better model real-world lighting conditions, we render images under a combination of a dominant point light (flash) and an environment map. We use the 49 environment maps used in [112], with random rotations. We sample the light source position from a Gaussian distribution centered at the camera to make the inference robust to differences in real-world mobile phones. We render linear images, though clamped to $(0,1)$ to mimic cameras with insufficient dynamic range. However, we still wish to reconstruct the full dynamic range of the SVBRDF parameters. To aid in this, we can render HDR images using in-our network rendering layer and compute reconstruction error w.r.t HDR ground truth images. In practice, this leads to unstable gradients in training; we mitigate this by applying a gamma of 2.2 and minor clamping to $(0,1.5)$ when computing the image reconstruction loss. We find that this, in combination with our L_2 losses on the SVBRDF parameters, allows us to hallucinate details from saturated images.

Training details

We use Adam optimizer [103] to train our network. We set $\beta_1 = 0.5$ when training the encoder and decoders and $\beta_1 = 0.9$ when training the classifier. The initial learning rate is set to be 10^{-4} for the encoder, 2×10^{-4} for the three decoders and 2×10^{-5} for the classifier. We cut down the learning rate by half in every two epochs. Since we find that the diffuse color and normal direction contribute much more to the final appearance, we first train their encoder-decoders for 15 epochs, then we fix the encoder and train the roughness decoder separately for 8

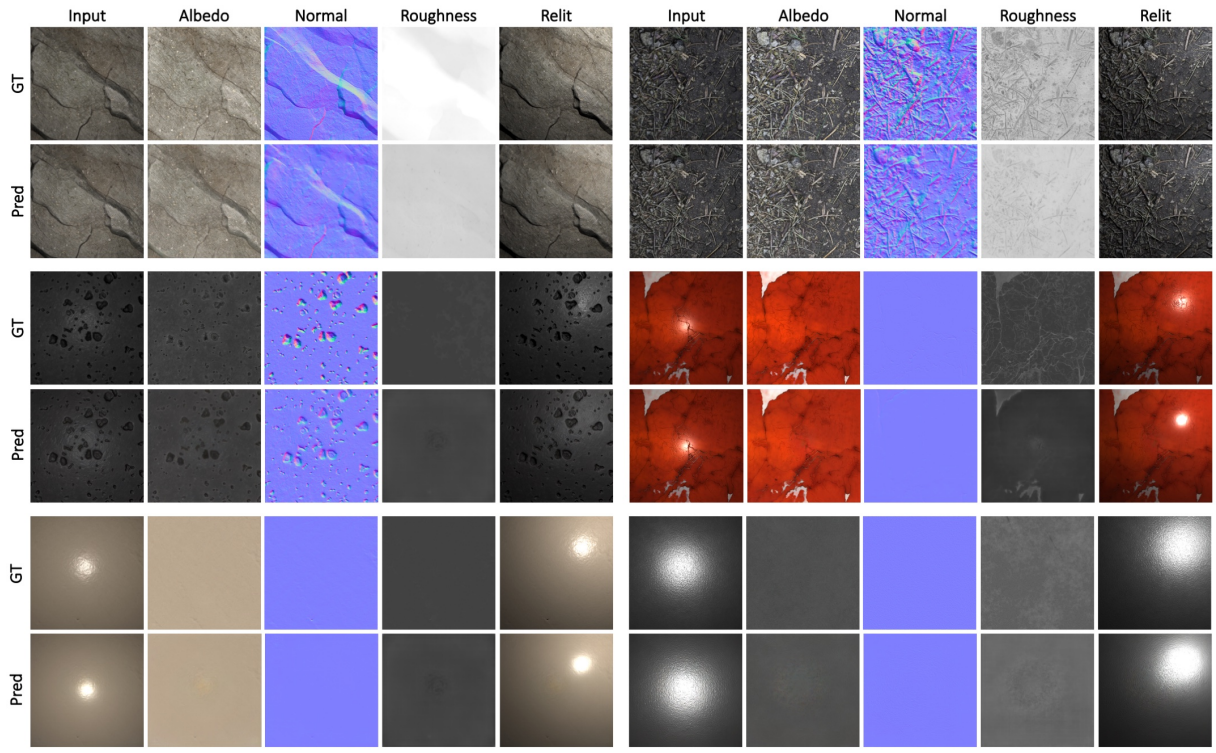


Figure 3.4. SVBRDF reconstruction results from our full method (`clsCRF-pt` in Table 3.3) on the test set. We compare the ground truth parameters with our reconstructions as well as renderings of these parameters under novel lighting. The accuracy of our renderings indicates the accuracy of our method.

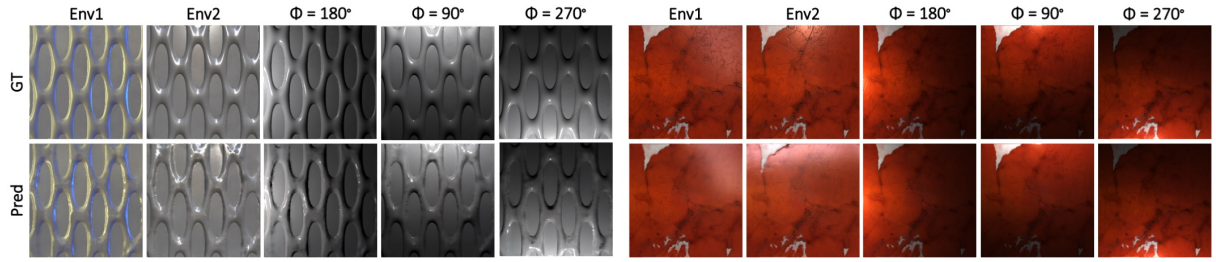


Figure 3.5. Materials estimated with our method and rendered under two environment lights and three point lights (placed on a unit sphere at $\theta = 50^\circ$ and various ϕ angles).

epochs. Next, we fix the network and train the parameters for the DCRFs, using Adam optimizer to update their coefficients.

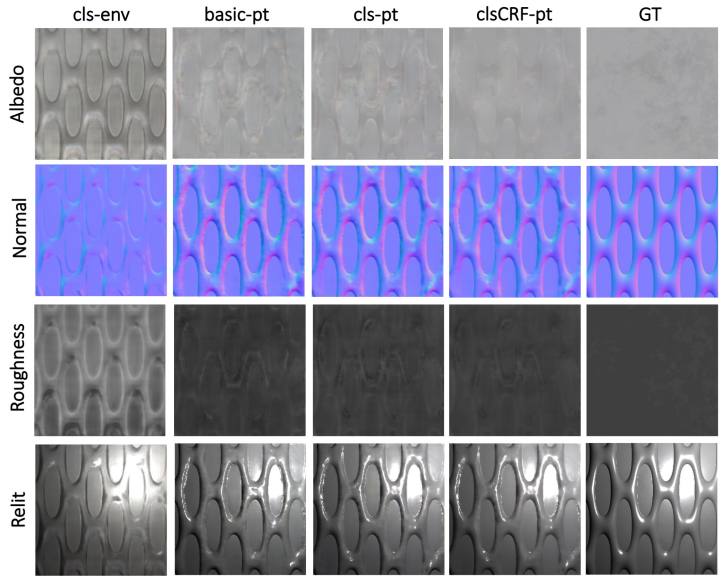


Figure 3.6. Qualitative comparison of BRDF reconstruction results of different variants of our network. The notation is the same as Table 3.3 and $-$`env` represents environment illumination.

3.5.1 Results on Synthetic Data

Qualitative results

Figure 3.4 shows results of our network on our synthetic test dataset. We can observe that spatially varying surface normals, diffuse albedo and roughness are recovered at high quality, which allows relighting under novel light source directions that are very different from the input. To further demonstrate our BRDF reconstruction quality, in Figure 3.5, we show relighting results under different environment maps and point lights at oblique angles. Note that our relighting results closely match the ground truth even under different lighting conditions; this indicates the accuracy of our reconstructions.

We next perform quantitative ablation studies to evaluate various components of our network design and study comparisons to prior work.

Effects of material classifier and DCRF

The ablation study summarized in Table 3.3 shows that adding the material classifier reduces the L_2 error for SVBRDF and normal estimation, as well as rendering error. This validates the intuition that the network can exploit the correlation between BRDF parameters

Table 3.3. Left to right: basic encoder-decoder, adding material classifier, adding DCRF and a pure material classifier. $-$`pt` indicates training and testing with dominant point and environment lighting.

Method	`basic-pt`	`cls-pt`	`clsCRF-pt`	`clsOnly-pt`
Albedo (e^{-3})	7.78	7.58	**7.42**	
Normal (e^{-2})	1.55	1.52	**1.50**	
Rough (e^{-2})	8.75	8.55	**8.53**	
Classify (%)		**73.65**	**73.65**	54.96

Table 3.4. BRDF reconstruction accuracy for different material types in our test set. Albedo-N is normalized diffuse albedo as in [112], that is, the average norm of each pixel will be 0.5.

		Albedo-N (e^{-4})	Normals (e^{-3})	Rough (e^{-2})
[112]	metal	91.8	27.2	–
	wood	35.9	11.2	–
	plastic	12.5	17.6	–
	Total	56.1	19.7	–
`cls-env`	metal	54.9	25.2	13.4
	wood	13.7	11.1	19.5
	plastic	7.96	14.2	25.3
	Total	30.9	18.1	18.0
`cls-pt`	metal	21.7	15.1	4.06
	wood	3.53	8.75	4.40
	plastic	1.64	9.10	7.24
	Total	11.3	11.7	4.83

and material type to produce better estimates. We also observe that training the classifier together with the BRDF reconstruction network results in a material classification error of 73.65%, which significantly improves over just our pure material classification network that achieves 54.96%. This indicates that features trained for BRDF estimation are also useful for material recognition. In our experiments, incorporating the classifier without using its output to fuse BRDF reconstruction results does not improve BRDF estimation. Figure 3.6 shows the reconstruction result on a sample where the classifier and the DCRF qualitatively improve the BRDF estimation, especially for the diffuse albedo.

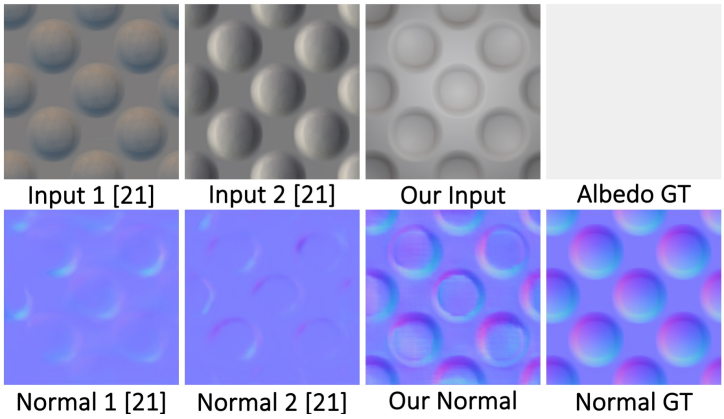


Figure 3.7. The first two inputs rendered under different environment maps are very different. Thus, the normals recovered using [112] are inaccurate. Our method uses point illumination (third input) which alleviates the problem, and produces better normals.

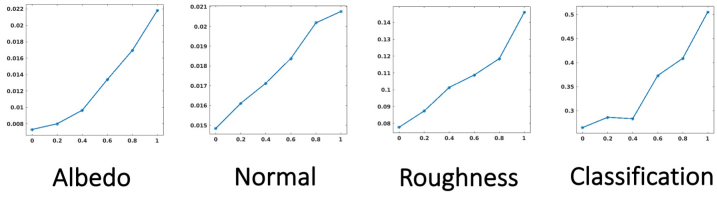


Figure 3.8. SVBRDF estimation errors with respect to relative intensities of environment against point light ranging from 0 to 0.8.

Effect of acquisition under point illumination

Next we evaluate the effect of using point illumination during acquisition. For this, we train and test two variants of our full network – one on images rendered under only environment illumination (`-env`) and another on images illuminated by a point light besides environment illumination (`-pt`). Results are in Table 3.4 with qualitative visualizations in Figure 3.6. The model from [112] in Table 3.4, which is trained for environment lighting, performs slightly worse than our environment lighting network `cls-env`. But our network trained and evaluated on point and environment lighting, `cls-pt`, easily outperforms both. We argue this is because a collocated point light creates more consistent illumination across training and test images, while also capturing higher frequency information. Figure 3.7 illustrates this: the appearance of the same material under different environment lighting can significantly vary and the network has to be invariant to this, limiting reconstruction quality.

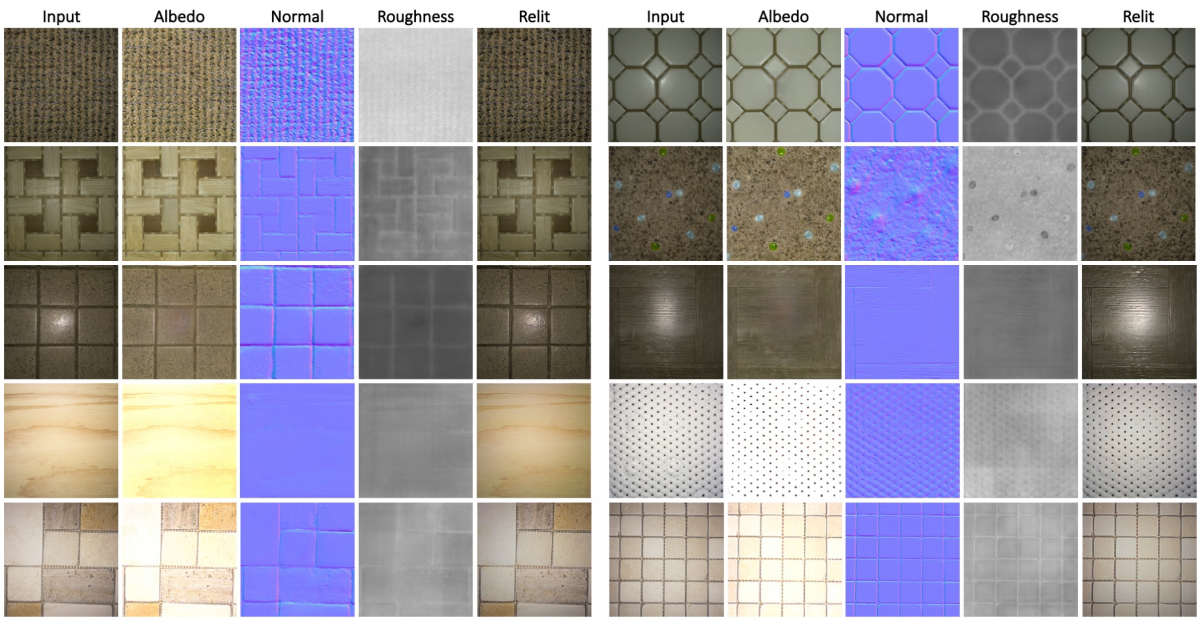


Figure 3.9. BRDF reconstruction results on real data. We tried different mobile devices to capture raw images using the Adobe LightRoom Mobile app. The input images in were captured using a Huawei P9 (first three rows), Google Tango (fourth row) and iPhone 6s (fifth row), all with a handheld mobile phone where the z-axis of camera was only approximately perpendicular to the sample surface.

Relative effects of flash and environment light intensities

In Figure 3.8, we train and test on a range of relative flash intensities. Note that as relative flash intensity decreases, errors increase, which justifies our use of flash light. Using flash and no-flash pairs can help remove environment lighting, but needs alignment of two images, which limits applicability.

3.5.2 Results on Real Data

Acquisition setup

To verify the generalizabity of our method to real data, we show results on real images captured with different mobile devices in both indoor and outdoor environments. We capture linear RAW images (with potentially clipped highlights) with the flash enabled, using the Adobe Lightroom Mobile app. The mobile phones were hand-held and the optical axis of the camera was only approximately perpendicular to the surfaces (see Figure 3.1).

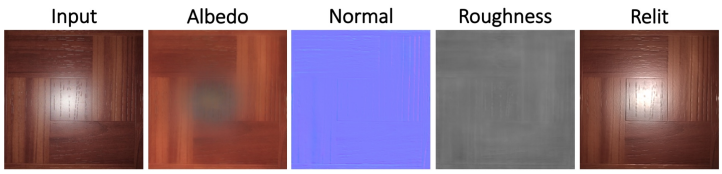


Figure 3.10. A failure case, due to incorrect material classification into *metal*, which causes the specularity to be over-smoothed.

Qualitative results with different mobile phones

Figure 3.9 presents SVBRDF and normal estimation results for real images captured with three different mobile devices: Huawei P9, Google Tango and iPhone 6s. We observe that even with a single image, our network successfully predicts the SVBRDF and normals, with images rendered using the predicted parameters appear very similar to the input. Also, the exact same network generalizes well to different mobile devices, which shows that our data augmentation successfully helps the network factor out variations across devices. For some materials with specular highlights, the network can hallucinate information lost due to saturation. The network can also reconstruct reasonable normals even for complex instances.

A failure case

In Figure 3.10, we show a failure case. Here, the material is misclassified as metal which causes the specular highlight in the center of image to be over-suppressed. In future work, we may address this with more robust material classification, potentially exploiting datasets like [19].

3.5.3 Further Comparisons with Prior Works

Comparison with two-shot BRDF method [6]

The two-shot method of [6] can only handle images with stationary texture while our method can reconstruct arbitrarily varying SVBRDFs. For a meaningful comparison, in Figure 3.12, we compare our method with [6] on a rendered stationary texture. We can see that even for this restrictive material type, the normal maps reconstructed by the two methods are quite similar, but the diffuse map reconstructed by our method is closer to ground truth. While [6]

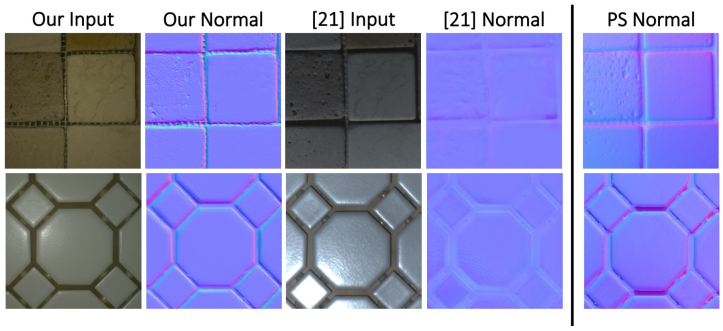


Figure 3.11. Comparison of normal maps using our method and [112], with photometric stereo as reference. Even with a lightweight acquisition system, our network predicts high quality normal maps.

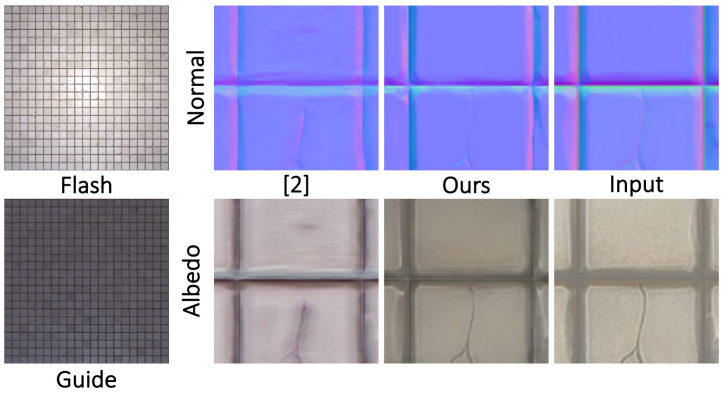


Figure 3.12. Comparison with [6], which requires two images, assumes stationary textures and takes over 6 hours (with GPU acceleration), yet our result is more accurate.

takes about 6 hours to reconstruct a patch of size 192×192, our method requires 2.4 seconds. The aligned flash and no-flash pair for [6] is not trivial to acquire (especially on mobile cameras with effects like rolling shutter), making our single image BRDF estimation more practical.

Comparison of normals with environment light and photometric stereo

In Figure 3.11, we compare our normal map and the results from a) [112] (from a single captured under environment lighting) and b) photometric stereo [81]. We observe that the normals reconstructed by our method are of higher quality than [112], with details comparable or sharper than photometric stereo.

3.6 Conclusion

We have proposed a framework for acquiring spatially-varying BRDF using a single mobile phone image. Our solution uses a CNN whose architecture is specifically designed to reflect various physical insights into the problem of BRDF estimation. We propose to use a dataset that is larger and better-suited to material estimation as compared to prior ones, as well as simple acquisition settings that are nevertheless effective for SVBRDF estimation. Our network generalizes very well to real data, obtaining high-quality results in unconstrained test environments. A key goal for our work is to take accurate material estimation from expensive and controlled lab setups, into the hands of non-expert users with consumer devices, thereby opening the doors to new applications. In the next chapter, we will take the next step of reconstruction SVBRDF with unknown shapes.

Chapter 3 is based on the material as it appears in European Conference on Computer Vision (ECCV), 2018 ("Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image", Zhengqin Li, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigator and author of this paper.

Chapter 4

Opaque Object Shape and SVBRDF Reconstruction from a Single Image

4.1 Introduction

Estimating the shape and reflectance properties of an object using a single image acquired "in-the-wild" is a long-standing challenge in computer vision and graphics, with applications ranging from 3D design to image editing to augmented reality. But the inherent ambiguity of the problem, whereby different combinations of shape, material and illumination might result in similar appearances, poses a significant hurdle. Consequently, early approaches have attempted to solve restricted sub-problems by imposing domain-specific priors on shape and/or reflectance [25, 13, 155]. Even with recent advances through deep learning based data-driven priors for inverse rendering problems, disentangling the complex factors of variation represented by arbitrary shape and spatially-varying bidirectional reflectance distribution function (SVBRDF) has, as yet, remained unsolved.

In this chapter, we take a step towards that goal by proposing a physically-motivated deep learning framework to estimate shape — represented as depth and surface normals — and SVBRDF — represented as diffuse albedo and specular roughness of the microfacet BRDF model (Eq. (2.6)) — from a single mobile phone image captured under largely uncontrolled conditions. This represents a significant advance over Chapter 3 that considers SVBRDF estimation from near-planar samples, or other recent works that estimate shape for Lambertian or homogeneous

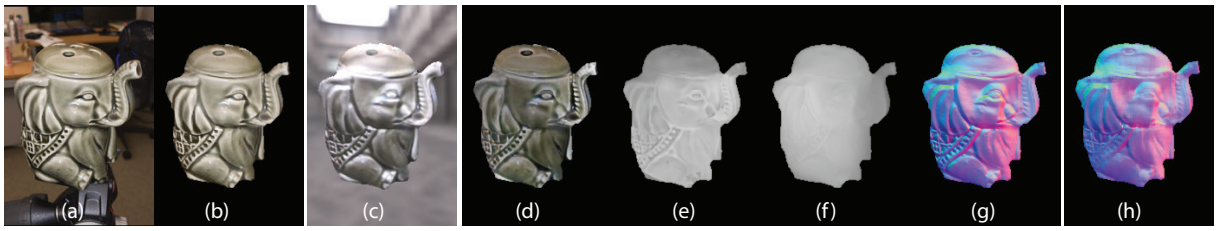


Figure 4.1. We propose a novel physically-motivated cascaded CNN architecture for recovering arbitrary shape and spatially-varying BRDF from a single mobile phone image. (a) Input image in unconstrained indoor environment with flash enabled. (b) Relighting output using estimated shape and SVBRDF. (c) Rendering output in novel illumination. (d–g) Diffuse albedo, roughness, depth and surface normals estimated using our framework. (h) Normals estimated using a single-stage network. Our cascade design leads to accurate outputs through global reasoning, iterative refinement and handling of global illumination.

materials [121, 64, 13]. The steep challenge of this goal requires a holistic approach that combines prudent image acquisition, a photorealistic large-scale synthetic training dataset, and novel physically-motivated networks that can efficiently handle this increased complexity.

In Chapter 3, we have demonstrated that a collocated source-sensor setup leads to advantages for material estimation, since higher frequencies for specular components are easily observed and distractors such as shadows are eliminated. Therefore, we use a mobile phone for imaging and mimic this setup by using the flash as illumination. Note that our images are captured under uncontrolled environment illumination, and not a dark room. Our only assumption is that the flash illumination is dominant, which is true for most scenarios.

Previous inverse rendering methods have utilized 3D shape repositories with homogeneous materials [121, 170, 186] or large-scale SVBRDFs with near-planar geometries [116, 54]. While we utilize the SVBRDF dataset in Chapter 3, meaningfully applying them to 3D models in a shape dataset is non-trivial. Moreover, category-specific biases in repositories such as ShapeNet [36] might mitigate the generalization ability of our learned model. To overcome these limitations, we procedurally generate random shapes by combining basic shape primitives on which the complex SVBRDFs from our dataset are mapped. We generate a large-scale dataset of $216,000$ images with global illumination that reflects the distribution of flash-illuminated images under an environment map.

Besides more descriptive datasets, disambiguating shape and spatially-varying material requires novel network architectures that can reason about appearance at multiple scales, for example, to understand both local shading and non-local shadowing and lighting variations, especially in the case of unknown, complex geometry. We demonstrate that this can be achieved through a cascade design; each stage of the cascade predicts shape and SVBRDF parameters, but these predictions and the error between images rendered with these estimates and the input image are passed as inputs to subsequent stages. This allows the network to imbibe this global feedback on the rendering error, while performing iterative refinement through the stages. In experiments, we demonstrate through quantitative analysis and qualitative visualizations that the cascade structure is crucial for accurate shape and SVBRDF estimation.

The forward rendering model is well-understood in computer graphics, and can be used to aid the inverse problem by using a fixed, in-network rendering layer to render the predicted shape and material parameters and impose a "reconstruction" loss during training, as shown in Chapter 3. However, tractable training requires efficient rendering layers, while the in-network rendering module in Chapter 3 only considers appearance under direct illumination. This is sufficient for planar surface but insufficient when dealing with arbitrary shapes. An important technical innovation of our network is a *global illumination* (GI) rendering layer that also accounts for interreflections.[1] While it is challenging to directly predict the entire indirect component of an input image, we posit that predicting the bounces of global illumination using a CNN is easier and maintains differentiability. Thus, our GI rendering is implemented as a physically-motivated cascade, where each stage predicts one subsequent bounce of global illumination. As a result, besides SVBRDF and shape, the individual bounces of global illumination are auxiliary outputs of our framework. A GI rendering layer also allows us to isolate the reconstruction error better, thereby providing more useful feedback to the cascade structure.

In summary, we make the following contributions:

[1] While it is possible to also consider shadows, global illumination is mainly manifested as interreflections in our inputs due to the collocated setup.

- The first approach to simultaneously recover unknown shape and SVBRDF using a single mobile phone image.
- A new large-scale dataset of images rendered with complex shapes and spatially-varying BRDF.
- A novel cascaded network architecture that allows for global reasoning and iterative refinement.
- A novel, physically-motivated global illumination rendering layer that provides more accurate reconstructions.

4.2 Related Work

Inverse rendering — the problem of reconstructing shape, reflectance, and lighting from a set of images — is an extensively studied problem in computer vision and graphics. Traditional approaches to this problem often rely on carefully designed acquisition systems to capture multiple images under highly calibrated conditions [53]. Significant research has also been done on the subproblems of the inverse rendering problem: e.g., photometric stereo methods that reconstruct shape assuming known reflectance and lighting [213], and BRDF acquisition methods that reconstruct material reflectance assuming known shape and lighting [130, 132]. While recent works have attempted to relax these assumptions and enable inverse rendering in the "wild", to the best of our knowledge, this paper is the first to estimate both *complex shape and spatially-varying non-Lambertian reflectance from a single image captured under largely uncontrolled settings*. In this section, we focus on work that addresses shape and material estimation from sparse images.

Shape and material estimation.

Shape from shading methods reconstruct shape from single images captured under calibrated illumination, though they usually assume Lambertian reflectance [91]. This has been extended to arbitrary shape and reflectance under known natural illumination [155]. Shape and reflectance can also be estimated from multiple images by using differential motion cues [31],

light field inputs [207, 113], or BRDF dictionaries [69, 82]. Recent works mitigate the challenge of shape recovery by using depth maps from a Kinect sensor as input for BRDF estimation [104, 215]. Other methods assume near-planar samples and use physics-based optimization to acquire spatially-varying BRDFs from sparse images captured under collocated illumination [6, 82, 172]. Yu et al. [233] assume known geometry to recover scene reflectance by modeling global illumination. Barron and Malik [13] recover shape and spatially-varying diffuse reflectance from a single image under unknown illumination by combining an inverse rendering formulation with hand-crafted priors on shape, reflectance and lighting. In contrast to these works, our deep learning approach recovers high-quality shape and spatially-varying reflectance from a single RGB image by combining a rendering layer with purely data-driven priors.

Deep learning for inverse rendering.

Recently, deep learning-based approaches have demonstrated promising results for several inverse rendering subproblems including estimating scene geometry [57, 11], material classes [19], illumination [78, 61, 64], and reflectance maps [170]. In contrast, our work tackles the joint problem of estimating shape and spatially-varying reflectance from just a single image.

In the context of reflectance capture, Aittala et al. [5] propose a neural style transfer approach to acquire stochastic SVBRDFs from images of near-planar samples under flash illumination. Similarly, Li et al. [112] acquire SVBRDFs from near-planar samples imaged under environment lighting, using a self-augmentation method to overcome the limitation of learning from a small dataset. Liu et al. [121] propose a CNN-based method, that incorporates an in-network rendering layer, to reconstruct a homegenous BRDF and shape (from one of four possible categories) from a single image under unknown environment illumination. [87] use deep networks to decompose images into intrinsic components like diffuse albedo, irradiance, specular and ambient occlusion, which are recombined to specify a render loss. We use a similar render loss, though our decomposition is physically-based. Meka et al. [136] recover homogeneous BRDF parameters of an arbitrary shape under environment lighting, and Chapter 3 and Deschaintre

et al. [54] leverage in-network rendering layers to reconstruct SVBRDFs from near-planar samples captured under flash illumination. Our work can be considered a generalization of all these methods — we handle a broader range of SVBRDFs and arbitrary shapes. This not only places greater demands on our network, but also necessitates the consideration of global illumination, leading to two key aspects of our architecture. First, we progressively refine shape and SVBRDF estimates through a novel cascade design. Second, while previous in-network rendering layers [121, 116, 54] only consider direct illumination, our global rendering layer accounts for indirect illumination too. This not only matches our inputs better, but is also the more physically accurate choice for real scenes with complex shapes. Further, the rendering error provided as input to our cascade stages improves estimation results, which is also possible only with a rendering layer that computes global illumination. Together, these components leads to state-of-the-art results on a significantly broader range of inputs.

Rendering layers in deep networks

Differentiable rendering layers have been used to aid in the task of learning inverse rendering for problems like face reconstruction [187, 200, 182] and material capture [121, 116, 54]. However, these methods make simplifying assumptions — usually Lambertian materials under distant direct lighting or planar surface with collocated point lighting — to make these layers tractable. We also use rendering to introduce information from varied lighting conditions, but in contrast to the above works, our rendering accounts for global illumination. Since analytical rendering of global illumination is challenging, we rely on network modules to predict bounces of global illumination. The idea of using a network to predict global illumination has also been adopted by [142], but no prior method has done this for inverse problems. Further, we use a physically meaningful network structure that divides global illumination into several bounces instead of directly predicting indirect lighting, which may lead to better and more interpretable results. A deep network is also used by [129] to compensate for global illumination in time-of-flight measurements, but they use a black box network for depth prediction while

we model global illumination explicitly. There is machinery to compensate for bounces in optimization-based methods [68], but they do not render in real-time and there is no obvious way to back-propagate gradients, making them unsuitable for our framework. We train a global illumination CNN to predict multiple bounces using data generated using a novel simulation-based strategy that renders random shapes with a large-scale SVBRDF dataset. The use of random shapes is important, since we aim to recover arbitrary geometry, unlike previous methods that might incorporate semantic category-level priors [170, 64, 121, 136, 36]. Besides higher accuracy in SVBRDF estimation, a collateral benefit of our novel rendering layer is that it can predict individual bounces of global illumination, in the same forward pass. These can be subsequently used for scene analysis tasks [145, 153].

Cascade networks

For prediction tasks that demand sub-pixel accuracy, prior works have considered cascade networks. For instance, convolutional pose machines [210] are devised to obtain large receptive fields for localizing human body joints, while other architectures such as deep pose [201] and stacked hourglass networks [147] also use cascades for multiscale refinement. Improved optical flow estimates are obtained by FlowNet 2.0 [86] using cascaded FlowNet modules that accept stage-wise brightness error as input. Similar to the above, we show that the cascade structure is effective for SVBRDF estimation. Uniquely, we demonstrate that our cascade is sufficient to recover high-quality shape and SVBRDF, while our global illumination prediction that enables rendering error as input to the cascade stages also yields advantages for SVBRDF estimation.

4.3 Method

The input to our method is a single image of an object (with a mask) captured under (dominant) flash and environment illumination. Reconstructing spatially-varying BRDF (SVBRDF) and shape, in such uncontrolled settings, is an extremely ill-posed problem. Inspired by the recent success of deep learning methods in computer vision and computer graphics, we handle

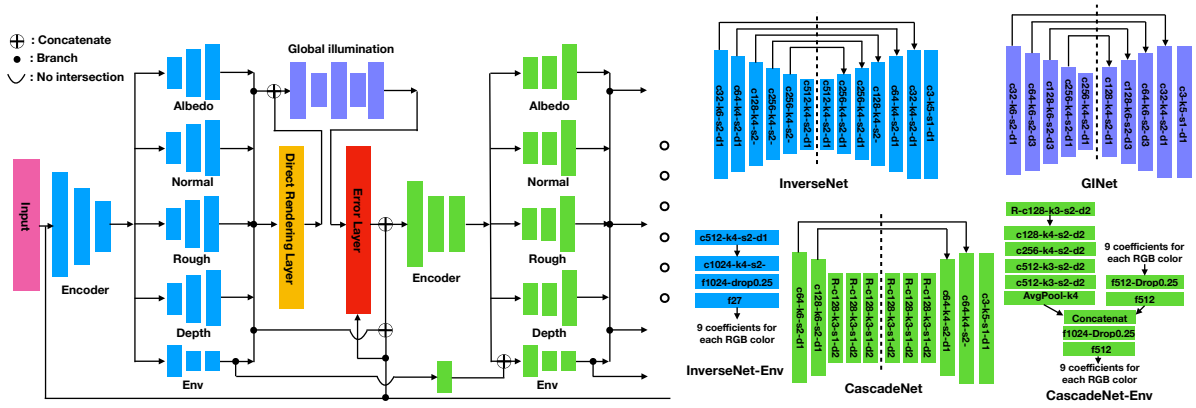


Figure 4.2. Right: Overall structure of our physically-motivated deep learning framework. Different colors specify different functions of the network (blue for initial estimation, green for refinement and purple for global illumination prediction). We use a cascade of encoder-decoder networks for global reasoning and iterative refinement. Different cascade levels do not share parameters since the input statistics at each stage and the refinements needed are different. Each cascade stage receives error feedback through the rendered output of the previous stage. Since we handle arbitrary shapes, our rendering layer models individual bounces of global illumination. **Left:** Details of hyperparameters in our physically-motivated network design. Here R represents a residual block [76]. $\mathtt{c}X_1-\mathtt{k}X_2-\mathtt{s}X_3-\mathtt{d}X_4$ represents a conv/deconv layer of output channel X_1, kernel size X_2, stride X_3 and dilation X_4. Our encoder has receptive fields large enough to model global light illumination, skip links are added since we aim to recover fine details and large kernels are used for global illumination prediction.

this problem by training a physically-motivated CNN specifically designed with intuition from physics-based methods. In this section, we will describe each component of our network. The overall framework is shown in Figure 4.2.

4.3.1 Basic Architecture

Our basic network architecture consists of a single encoder and four decoders for different shape and SVBRDF parameters: diffuse albedo (**A**), specular roughness (**R**), surface normal (**N**), and depth (**D**).[2] For simplicity, we start by considering the input to be an image, $\mathbf{I}^p$, of an object illuminated by a dominant point light source collocated with the camera (we consider additional environment illumination in Section 4.3.3). We manually create a mask, **M**, that we

[2] A specular albedo may be considered too, but we found it sufficient to consider just roughness to mimic most real-world appearances.

stack with the image to form a four channel input for the encoder. A light source collocated with the camera has the advantages of removing cast shadows, simplifying the lighting conditions and easing observation of high frequency specularities, which are crucial for solving the inverse rendering problem. In our experiments, such input data is easily acquired using a mobile phone with the flash light enabled. Similar to Chapter 3, our four decoders share features extracted from the same encoder. The intuition behind this choice is that different shape and SVBRDF parameters are closely correlated, thus, sharing features can greatly reduce the size of the network and alleviate over-fitting. Let **InverseNet**$(\cdot)$ be the basic network architecture consisting of the encoder-decoder block (shown in blue in Figure 4.2). Then the initial predicted shape and SVBRDF estimates are given by:

$$\mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D} = \textbf{InverseNet}(\mathbf{I}^p, \mathbf{M}). \tag{4.1}$$

4.3.2 Global Illumination Rendering Layer

Prior works on material capture or photometric stereo usually assume that the influence of inter-reflections can be neglected, or consider near-planar samples where its effects are not strong, as discussed in Chapter 3. However, that may not be the case for our setup, since we consider complex shape with potentially glossy reflectance. Failing to model global illumination for our problem can result in color bleeding and flattened normal artifacts. We initially considered in-network global illumination rendering during training, but found it time-consuming and not feasible for a large dataset. Instead we propose using CNNs to approximate global illumination. CNNs can capture the highly non-linear operations that global illumination manifests. In addition, they have the advantage of being differentiable and fast to evaluate.

In particular, we use a series of CNNs, each of which predict individual bounces of the rendered image. Let $\textbf{GINet}_n$ be the n-bounce CNN. This network is trained to takes the $(n-1)$-bounce image under point light illumination, $\mathbf{I}^p_{n-1}$, and the shape and SVBRDF parameters, and

render the n-bounce image, $\mathbf{I}_n^p$, as:

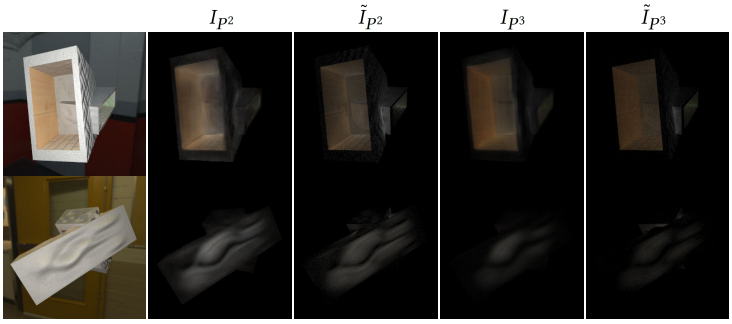
$$\mathbf{I}_n^p = \mathbf{GINet}_n(\mathbf{I}_{n-1}^p, \mathbf{M}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D}) \tag{4.2}$$

We use an analytical rendering layer to compute the direct illumination, i.e., first bounce image, $\tilde{\mathbf{I}}_1^p$, given the predicted shape and SVBRDF parameters. Then we use two CNNs, $\mathbf{GINet}_2(\cdot)$ and $\mathbf{GINet}_3(\cdot)$, to predict the second and third bounces, $\mathbf{I}_2^p$ and $\mathbf{I}_3^p$ respectively. The output, $\mathbf{I}_g^p$, of our full global illumination rendering layer (shown in purple in Figure 4.2) sums all the bounce images as:

$$\begin{aligned} \mathbf{I}_2^p &= \mathbf{GINet}_2(\mathbf{I}_1^p, \mathbf{M}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D}), \\ \mathbf{I}_3^p &= \mathbf{GINet}_3(\mathbf{I}_2^p, \mathbf{M}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D}), \\ \mathbf{I}_g^p &= \mathbf{I}_1^p + \mathbf{I}_2^p + \mathbf{I}_3^p. \end{aligned} \tag{4.3}$$

As illustrated in Figure 4.3, most of the image intensity is contained within three bounces, and so we only predict these, ignoring subsequent bounces. Also in Figure 4.3, we show second and the third bounce images predicted by our network. We observe that even for objects with very concave shape and highly glossy material, we can still generate rendering outputs that closely match the ground truth.

Note that a CNN-based approach like ours only approximates true global illumination. It operates in image space and does not explicitly model interreflections from surface points that are not visible to the camera. However, our training data does include interreflections from invisible surfaces and our collocated setup causes interreflections from visible regions to dominate. In practice, we have found the network to be sufficiently accurate for inverse rendering. Compared with the traditional radiosity method [47], our network-based global illumination prediction has the advantage of being fast, differentiable and able to approximate reflections from invisible surfaces. However, it is an approximation, since we do not have precise geometry, form factors



	1^{st}	2^{nd}	3^{rd}	others
Energy ratio	95.83%	3.08%	0.89%	0.20 %

Figure 4.3. Global illumination prediction results. From left to the right are input images, the predicted second bounce images, the ground truth second bounce images, the predicted third bounce images and the ground truth third bounce images. Even for complex shapes with glossy material, the predictions of our network are close to the ground truth. On the bottom, we show the ratio between the average energy of separate bounces and the images illuminated by a point light source across the test dataset.

or material (albedo) properties, as in conventional radiosity algorithms.

4.3.3 Environment Map Prediction

Although we use a dominant flash light, our images are also illuminated by unknown environment illumination. This environment illumination can significantly affect the appearance of globally illuminated complex shapes. This requires us to estimate the environment illumination and account for it in our rendering networks. To do so, we approximate environment lighting with low-frequency spherical harmonics (SH), and add another branch to our encoder-decoder structure to predict the first nine SH coefficents for each color channel. We observe that the background image provides important context information for the network to determine environment lighting. So, unlike the point light source case, we add the image with background as the third image to the input. Let $\mathbf{E}$ be environment lighting, $\mathbf{I}^{pe}$ be the image of the object under both point and environmental lighting and $\mathbf{M} \odot \mathbf{I}^{pe}$ be its masked version. With some

abuse of notation, now our shape and SVBRDF parameters are computed using

$$\mathbf{A},\mathbf{N},\mathbf{R},\mathbf{D},\mathbf{E} = \textbf{InverseNet}(\mathbf{I}^{pe},\mathbf{M}\odot\mathbf{I}^{pe},\mathbf{M}). \tag{4.4}$$

Since now the input image is captured under environment illumination and the flash light source, we modify our rendering layer to account for this. We follow the method of [167] to render an image of the object, $\mathbf{I}_e$, using the estimated spherical harmonics illumination. This only considers the Lambertian shading and ignores high-frequency specular effects. In practice, this is sufficient because most high-frequency effects are observed under flash illumination, and our experiments show that this simple approximation suffices for achieving accurate BRDF reconstruction. Now the output of the global illumination rendering layer (in place of Equation 4.3) is given by:

$$\mathbf{I}_g^{pe} = \mathbf{I}_1^p + \mathbf{I}_2^p + \mathbf{I}_3^p + \mathbf{I}^e. \tag{4.5}$$

4.3.4 Cascade Structure

While a single encoder-decoder leads to good results for SVBRDF estimation with near-planar samples in Chapter 3, it does not suffice when considering arbitrary shapes. This can be attributed to the increased complexity of the problem and a need for more global reasoning. We propose a cascade structure that achieves these aims by using iterative refinement and feedback to allow the network to reason about differences between the image rendered with the predicted parameters and the input image.

Let $\textbf{CascadeNet}_n$ be stage n of the cascade network. Each stage has the same single architecture as **InverseNet**. Let the shape, reflectance and lighting parameters of cascade stage n be $\mathbf{A}_n$, $\mathbf{N}_n$, $\mathbf{R}_n$, $\mathbf{D}_n$ and $\mathbf{E}_n$, and the result of rendering these parameters (using the global illumination rendering network) be $\mathbf{I}_{g,n}^{pe}$. Each cascade stage refines the predictions of the

previous stage as:

$$\mathbf{Err}_{n-1} = \mathbf{M} \odot \mathbf{I}^{pe} - \mathbf{I}^{pe}_{g,n-1} \tag{4.6}$$

$$\mathbf{A}_n, \mathbf{N}_n, \mathbf{R}_n, \mathbf{D}_n, \mathbf{E}_n = \mathbf{CascadeNet}_n(\mathbf{I}^{pe}, \mathbf{M} \odot \mathbf{I}^{pe}, \mathbf{M}, \mathbf{A}_{n-1}, \mathbf{N}_{n-1}, \mathbf{R}_{n-1}, \mathbf{D}_{n-1}, \mathbf{Err}_{n-1}) \tag{4.7}$$

The inputs to each cascade stage are the input image, the shape, SVBRDF, and lighting predictions from the previous stage, and the rendering error associated with these previous predictions (with respect to the input image). This allows each cascade stage to refine the predictions by reasoning about the rendering error from the previous stage. Note that this is possible only because of our network design that models global illumination and environment lighting.

4.3.5 Training Details

Training Data:

To the best of our knowledge, there is no existing dataset of objects with arbitrary shape rendered with complex SVBDRF. Complex SVBRDF datasets used in Chapter 3 assume near-planar surfaces, and rich shape datasets like ShapeNet [36] have simple homogeneous BRDFs. Thus, we generate our own synthetic dataset by procedurally adding shapes to build a complex scene. Similar to [226], we first generate primitive shapes (cube, ellipsoid, cylinder, box and L-shape) and then add a randomly generated height map to make them more complex and diverse. We build scenes by randomly sampling 1 to 5 shapes and combining them. We create 3600 scenes, using 3000 for training and 600 for testing.

We use SVBDRFs from the Adobe Stock material dataset[3], which contains 694 complex SVBRDFs spanning a large range of material types. Each SVBRDF is comprised of 4K texture maps for diffuse albedo, specular roughness, and surface normals. For data augmentation, we

[3]https://stock.adobe.com/3d-assets

randomly crop, rotate and scale the texture maps to create different size patches and then resize all the patches to the same size of 256×256. We create 270 patches for each material and use these patches as the materials to render dataset. We use the physically motivated microfacet BRDF model defined in (2.6). We remove the 6 transparent materials and use the remaining 688 materials. We classify the materials into 8 categories according to their reflectance properties as shown in Figure 3.2 and proportionally sample 588 materials for training and 100 for testing. For environment maps, we use the Laval Indoor HDR dataset [61] containing 2144 environmental maps of indoor scenes, of which we use 1500 to render the training dataset and 644 for the test dataset.

We use Optix for GPU-accelerated rendering, based on path tracing with multiple importance sampling. We render with 400 samples per-pixel for point light source illumination and 625 samples per-pixel when the environment map is also included. The average rendering time is less than 2 seconds. For each scene, we sample 12 viewing directions, 5 groups of different SVBDRFs and one environment map. When rendered with both point and environment lighting, we scale the environment map by 0.5, to keep the average ratio between image intensities rendered with only environment map and with point light to be 0.09285. This matches the statistics of images captured using mobile phones in real indoor environments.

Network Design:

Our design makes several choices to reflect the physical structure of the problem. We use the U-net architecture [176] for **InverseNet**. To model the global fall-off of the point light source, it is necessary to have large receptive fields. Thus, each encoder has 6 convolutional layers with stride 2, so that each pixel of the output can be influenced by the whole image. For the SVBDRF parameters, we use transposed convolutions for decoding and add skip links to recover greater details. For environment map estimation, we pass the highest level of feature extracted from the encoder through two fully connected layers to regress the 9 spherical harmonics coefficients. Each **CascadeNet** stage uses 6 residual blocks — 3 blocks for the encoder and 3 separate blocks

for each decoder. We use dilated convolutions with a factor of 2 in the residual block to increase the receptive field. We feed environment lighting predictions into the next cascade stage by passing the nine SH coefficients through a fully connected layer and concatenate them with the feature extracted from the encoder. We also use the U-net structure with skip-links for **GINet**. To predict global illumination, the network must capture long range dependencies. Thus, we use a convolutional layer with large kernel of size 6, combined with dilation by a factor of 2. The network architecture of each component is shown on the right side of Figure 4.2.

Loss function:

We have the same loss function for both **InverseNet** and each **CascadeNet** stage. For diffuse albedo, normal, roughness and environment illumination SH coefficients, we use the L_2 loss for supervision. Since the range of depths is larger than that of other BRDF parameters, we use an inverse transformation to project the depth map into a fixed range. Let $\tilde{\mathbf{D}}_i$ be the initial output of depth prediction network of pixel i; the final depth $\mathbf{D}_i$ is given by

$$\mathbf{D}_i = \frac{1}{\sigma \cdot (\tilde{\mathbf{D}}_i + 1) + \varepsilon}. \tag{4.8}$$

We set $\sigma = 0.4$ and $\varepsilon = 0.25$, and use L_2 loss to supervise $\mathbf{D}_i$. Finally, we add a reconstruction loss based on the L_2 distance between the image rendered with predicted and ground truth parameters. Let $\mathscr{L}_a$, $\mathscr{L}_n$, $\mathscr{L}_r$, $\mathscr{L}_d$, $\mathscr{L}_{env}$ and $\mathscr{L}_{rec}$ be the L_2 losses for diffuse albedo, normal, roughness, depth, environment map and image reconstruction, respectively. The loss function of our network is:

$$\mathscr{L} = k_a\mathscr{L}_a + k_n\mathscr{L}_n + k_r\mathscr{L}_r + k_d\mathscr{L}_d + k_{env}\mathscr{L}_{env} + k_{rec}\mathscr{L}_{rec}, \tag{4.9}$$

where $k_a = k_n = k_{rec} = 1$, $k_r = k_d = 0.5$ and $k_{env} = 0.1$ are parameters chosen empirically.

Training Strategies:

Training multiple cascade structures is difficult since the enhanced network depth may lead to vanishing gradients and covariate shift, preventing convergence to a good local minimum. Further, batch sizes will need to be small when training all stages together, which can cause instability. Thus, instead of training the whole network end-to-end, we sequentially train each stage of the cascade. This allow us to use a relatively large batch size of 16. We use Adam optimizer, with a learning rate of 10^{-4} for the encoder and 4×10^{-4} for the decoders. We decrease the learning rate by half after every two epochs. The three stages are trained for 15, 8 and 6 epochs, respectively. We use two **CascadeNet** stages and train **InverseNet** and **CascadeNet**$_1$ with 2500 shapes and add 500 shapes to train **CascadeNet**$_2$.

GINet is trained prior to the BRDF prediction network, then held fixed and only used for the rendering layer when training the network for shape and SVBRDF estimation. We use Optix to render images with separate bounces and use them for direct supervision. We train for 15 epochs, with an initial learning rate of 2×10^{-4} and reduce it by half every two epochs.

4.4 Experiments

We first demonstrate the effectiveness of each design choice in our network architecture through detailed comparisons on both synthetic and real datasets. Next, we compare with previous methods for shape and material estimation to highlight the highly accurate shape and SVBRDF recovered by our framework. The accompanying video demonstrates our high-quality joint shape and material reconstruction through photorealistic editing applications, including relighting and novel view synthesis.

Ablation study on synthetic data

We first justify the necessity of rendering a novel large-scale dataset with global illumination for shape and SVBRDF estimation. We train **InverseNet** on images rendered with direct illumination and test on images with global illumination. Column $\mathtt{Im}_{\mathtt{d}}^{\mathtt{p}}\mathtt{-C0}$ (trained on

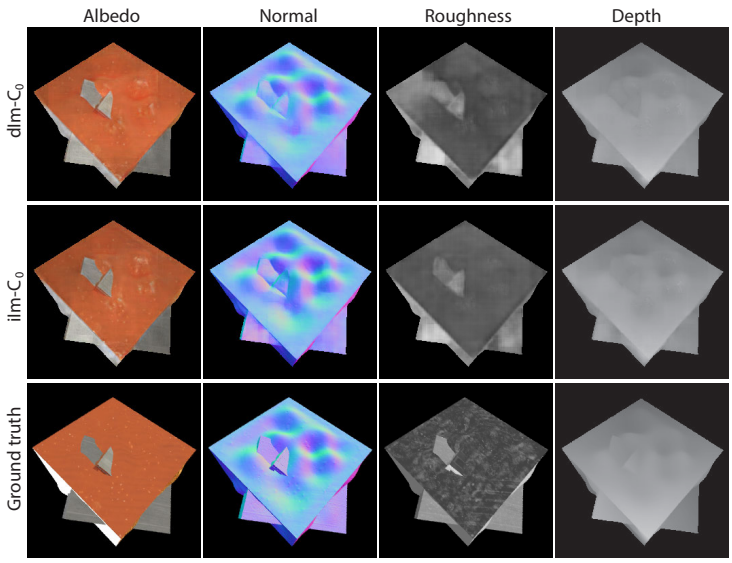


Figure 4.4. Comparison of SVBRDF and depth outputs of two networks, trained on directly illuminated (top) and globally illuminated images (middle), when evaluated on an input with global illumination. Not considering indirect lighting during training leads to flatter normals and brighter albedo.

Table 4.1. Quantitative comparison on images rendered only with point light. $\mathtt{Im}_{\mathtt{d}}^{\mathtt{p}}$ refers to input images rendered with direct lighting only, while $\mathtt{Im}_{\mathtt{g}}^{\mathtt{p}}$ means the input images are rendered with global illumination.

	$\mathtt{Im}_{\mathtt{d}}^{\mathtt{p}}-\mathtt{C0}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{p}}-\mathtt{C0}$
Albedo(10^{-2})	5.911	5.703
Normal(10^{-2})	4.814	4.475
Roughness(10^{-1})	1.974	1.966
Depth(10^{-1})	1.842	1.772

images with direct point illumination with no cascade.) in Table 4.1 reports the obtained errors, which are clearly larger than those in column $\mathtt{Im}_{\mathtt{g}}^{\mathtt{p}}-\mathtt{C0}$ for the same network trained on images with point lighting and global illumination. Thus, global illumination has a significant impact on depth and SVBRDF estimation. The qualitative comparison in Figure 4.4 shows that the network trained with direct lighting only predicts brighter diffuse albedo and flattened normals, when evaluated on images with indirect lighting. This also matches intuition on the behavior of inter-reflections [144].

Next we demonstrate that context information is important for the network to reconstruct

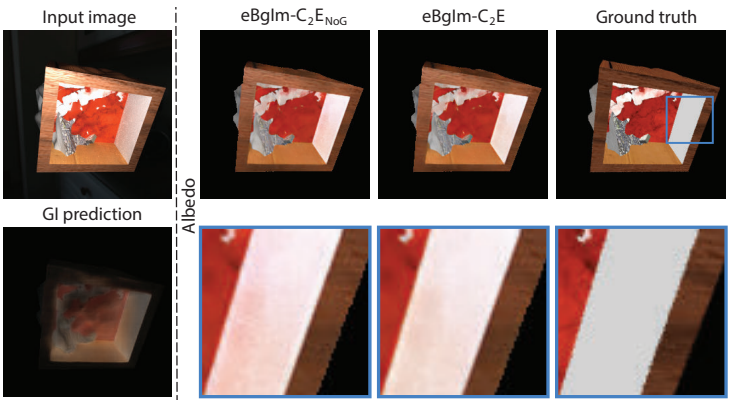


Figure 4.5. For an input image with strong indirect lighting (top left), a network trained without global illumination for the rendering layer (second column) retains more color bleeding artifacts in the estimated diffuse albedo, than one trained with global illumination (third column). The bottom left figure shows the net global illumination estimated by the final network.

shape and BRDF under environment lighting. We train two variants of our basic network, one with masked image input, $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{C}_0$, and the other with both masked and original image as input, $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_0$. Quantitative comparisons in the first two columns of Table 4.2 show that predictions for all BRDF parameters improve when background is included.

To test the effectiveness of cascade structure, we first add one layer of cascade to our basic network. We try two variants of cascade network. For the black-box cascade ($\mathtt{C}_1$), we stack the input image and the predicted BRDF parameters and send them to the next stage of the cascade. For the cascade network with error feedback ($\mathtt{C}_1\mathtt{Er}$), we also send an error map as input by comparing the output of our global illumination rendering layer with the input. The quantitative numbers (third and fourth column of Table 4.2) suggest that having the error feedback improves BRDF reconstruction. We then add another cascade stage with error feedback, which yields even more accurate BRDF estimation ($\mathtt{C}_2\mathtt{Er}$) that we deem the final output. Figure 4.9 shows the visual quality of BRDF estimation from different stages of the cascade network. We observe that for both synthetic and real data, the cascade reduces noise and artifacts. The final rendered result using the BRDF parameters predicted by the second level of the cascade is very similar to the input image, as shown in Figure 4.9 using both the environment map estimated by the network and a novel environment map.

Table 4.2. Quantitative comparisons L_2 errors illustrating the influence of various network choices. All input images $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}$ here are illuminated by both point light source and environmental lighting pe and rendered with global illumination g. bg means the images without masking the background are added as an input. $\mathtt{C}_{\mathtt{n}}$ shows the level of cascade refinement, where $\mathtt{C}_{\mathtt{0}}$ means we use our basic InverseNet without any refinement. Er behind $\mathtt{C}_{\mathtt{n}}$ means we also send the error maps by comparing the images rendered with the estimated BRDFs and the inputs to the cascade refinement networks. The subscript NoE and NoG in the last two columns means that when computing the error maps, we do not consider the influence of environmental lighting and global illumination respectively. Here, $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}$ is the error obtained with our final two-cascade architecture with global illumination and error feedback.

	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{C}_{\mathtt{0}}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{0}}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{1}}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{1}}\mathtt{Er}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}_{\mathtt{NoE}}$	$\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}_{\mathtt{NoG}}$
Albedo(10^{-2})	6.089	5.670	5.150	5.132	**4.868**	4.900	4.880
Normal(10^{-2})	4.727	4.580	3.929	3.907	**3.822**	3.830	**3.822**
Roughness(10^{-1})	2.207	2.064	2.004	2.011	**1.943**	1.948	1.947
Depth(10^{-2})	1.945	1.871	1.631	1.624	**1.505**	1.512	1.511
Bounce 1(10^{-3})	3.526	3.291	2.190	2.046	**1.637**	1.643	1.643
Bounce 2(10^{-4})	2.88	2.76	2.47	2.47	**2.45**	2.45	2.46
Bounce 3(10^{-5})	6.6	6.4	5.9	5.9 .	**5.8**	**5.8**	**5.8**

Next, we analyze the effect of the global illumination rendering network. We train two new variants of our global illumination rendering layer for the second cascade stage. For $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}_{\mathtt{NoG}}$, the rendering layer does not consider global illumination so that the error feedback is computed by subtracting the sum of $\tilde{\mathbf{I}}_1^p$ and $\tilde{\mathbf{I}}^e$ from the input $\mathbf{M}\odot\mathbf{I}^{pe}$, i.e., $\mathbf{I}_n^{pe}=\mathbf{I}_{1,n}^p+\mathbf{I}_n^e$. Similarly, for $\mathtt{Im}_{\mathtt{g}}^{\mathtt{pe}}-\mathtt{bg}-\mathtt{C}_{\mathtt{2}}\mathtt{Er}_{\mathtt{NoE}}$, we remove the environmental map component of the global illumination rendering layer. The error feedback for the cascade network is now computed using $\mathbf{I}_n^{pe}=\mathbf{I}_{1,n}^p+\mathbf{I}_{2,n}^p+\mathbf{I}_{3,n}^p$. Table 4.2 shows that our full version of rendering layer performs the best. The differences are measurable but subtle, since the remaining impact of environment lighting and global illumination for the second stage is small. To better understand the behavior, we show a qualitative example with global illumination in Figure 4.5. We observe that the global illumination rendering layer alleviates color bleeding artifacts.

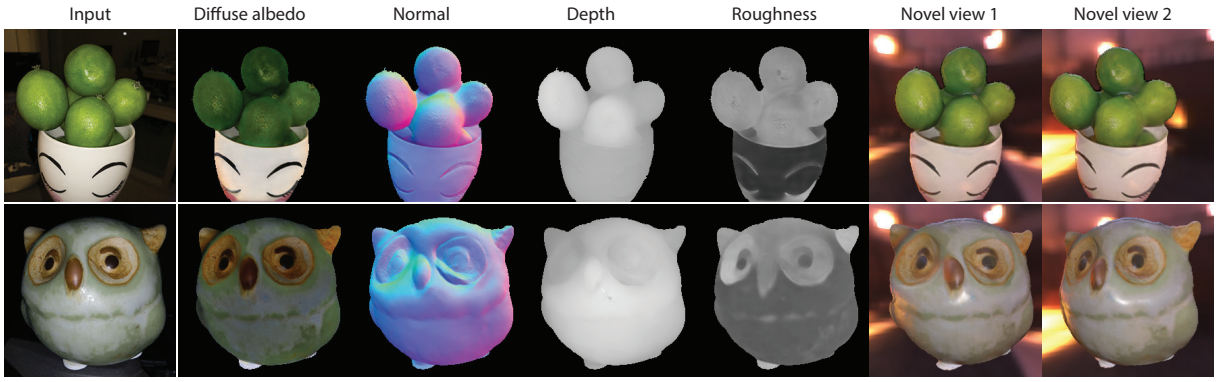


Figure 4.6. Results rendered from novel views. We show the input image, the estimated shape and BRDF parameters and the rendered output under an environment map from two novel views. We observe high fidelity rendered images, as well as high quality recovery of shape and spatially-varying BRDF.

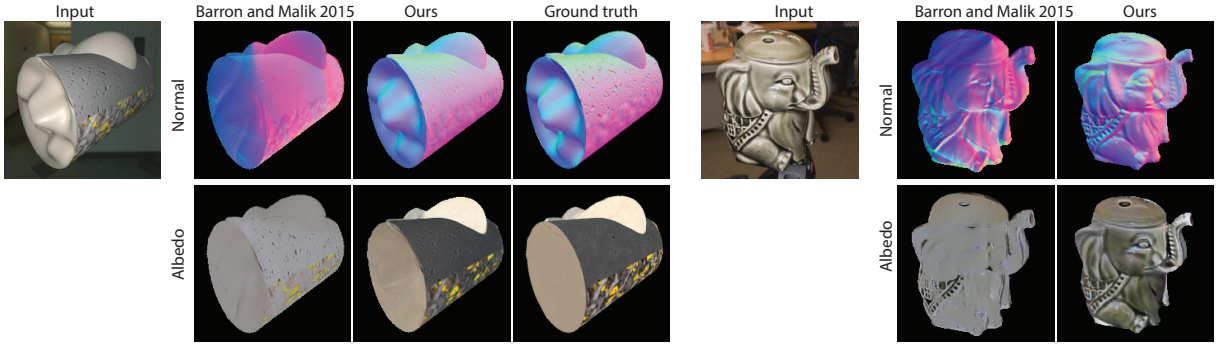


Figure 4.7. Comparison with SIRFS [13]. Our method accurately estimates the shape and diffuse color, even in regions with specularity. In contrast, because of the complex shape and materials of these objects, SIRFS, which assumes Lambertian reflectance, produces very inaccurate estimates.

Generalization to real data

We demonstrate our method on several real objects in Figures 4.10 and 4.6. All images are captured in indoor scenes using an iPhone 10 with the flash enabled. We use the Adobe Lightroom app to capture linear images and manually create the segmentation mask. For all the examples, our rendered results closely match the input. Figures 4.10 and 4.6 also show our predicted BRDF parameters can be used to render realistic images under new environment lighting and camera pose. This demonstrates that our estimates of the surface normal and spatially varying roughness are of high enough quality to render realistic specular effects of real objects under novel illumination and viewing directions.

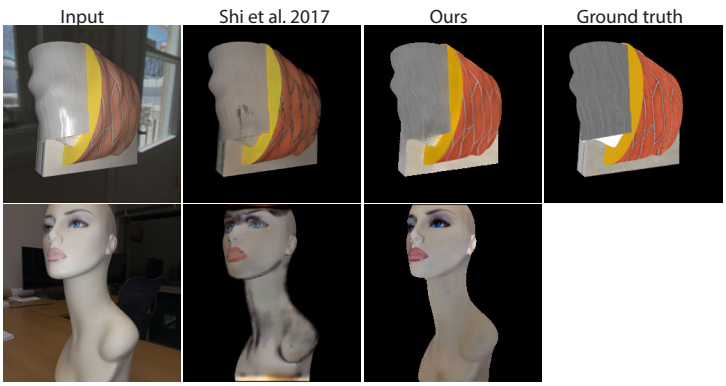


Figure 4.8. Comparison with [186]. While Shi et al. train to handle non-Lambertian reflectance, the accuracy and visual quality of our diffuse albedo is significantly higher on both synthetic (top) and real data (bottom).

Comparisons with previous methods

Since we are not aware of prior works that can use a single image for spatially varying BRDF and shape estimation, our comparisons are to more restricted methods for shape and material estimation, or to intrinsic image decomposition methods. We first compare with a model-based method. SIRFS [13], which jointly reconstructs shape and diffuse color. Figure 4.7 compares the diffuse albedo and normal estimated using SIRFS with those obtained by our framework, on both real and synthetic data. In both cases, our estimates are significantly better. Notice that SIRFS tends to over-smooth both the diffuse color and the normal due to a handcrafted regularization. In contrast, our method successfully recovers high-frequency details for both diffuse albedo and surface normals, even in specular and shadowed regions.

We also compare with the recent deep learning-based intrinsic image decomposition method of [186], which is trained to separate diffuse and specular components from a single image of a ShapeNet object [36], rendered under the assumption of a parametric homogeneous BRDF. We compare to their diffuse albedo prediction in Figure 4.8. Our physically-motivated framework can better preserve occlusion boundaries and recover accurate diffuse color even in specular regions. Our method also yields qualitatively superior results on real data.

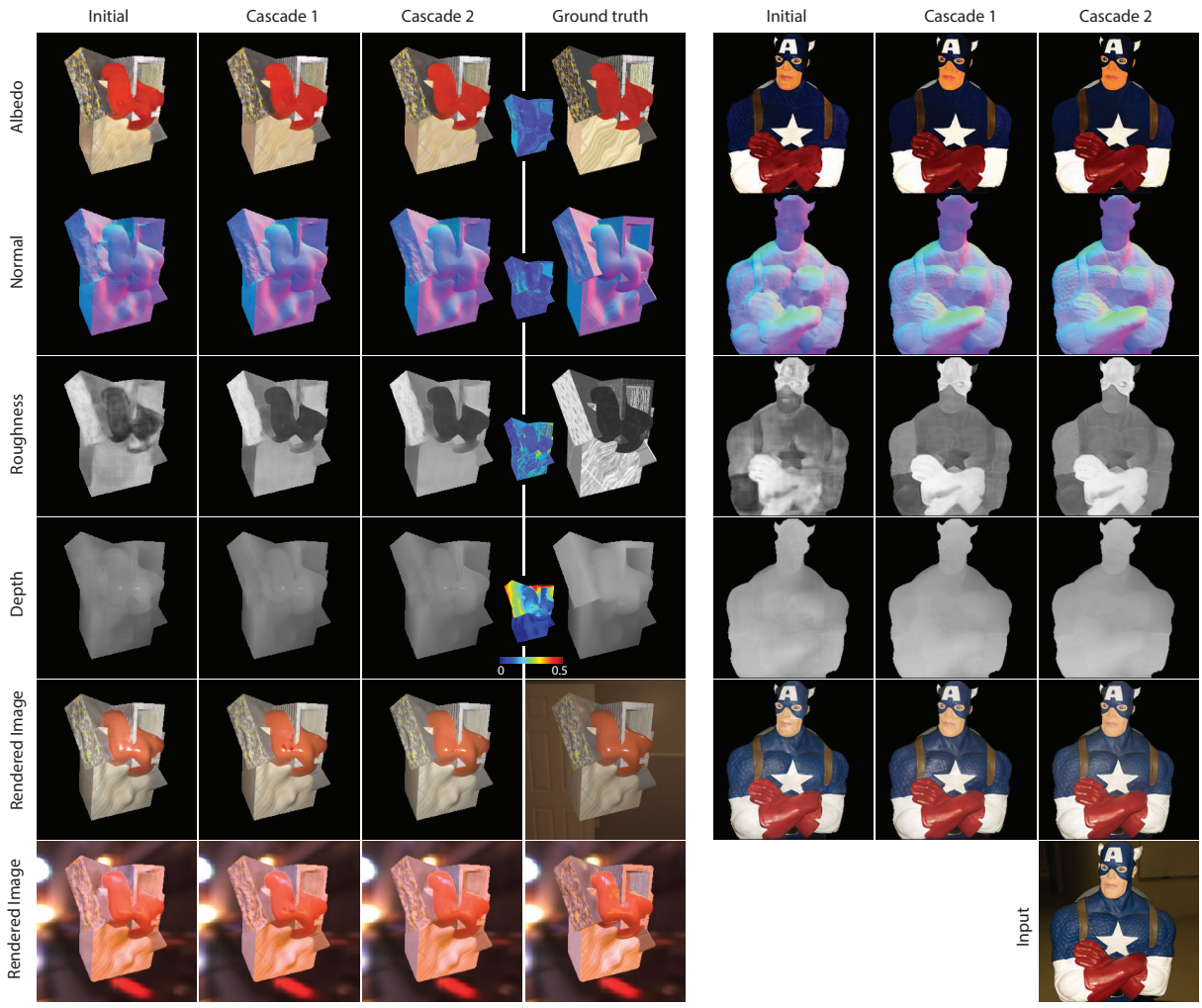


Figure 4.9. Effect of our cascaded design, illustrated for synthetic (left) and real data (right). We visualize the absolute error for the BRDF parameters in the third column except the depth error. The depth error is normalized so that the range of ground-truth depth is 1.

Limitations

A few challenges remain unaddressed. Our network does not explicitly handle improperly exposed images. For example, saturations from the flash may cause the specular highlight to be baked into the diffuse color (such as the orange in the third row of Figure 4.10). This problem might be solved by adding more training data and using more aggressive data augmentation. As discussed previously, long-range interactions might not be sufficiently modeled in our image-space CNN, which may limit its ability to correctly handle interreflections. We find spatially varying roughness prediction to be a challenging problem. The presence of specular highlights is

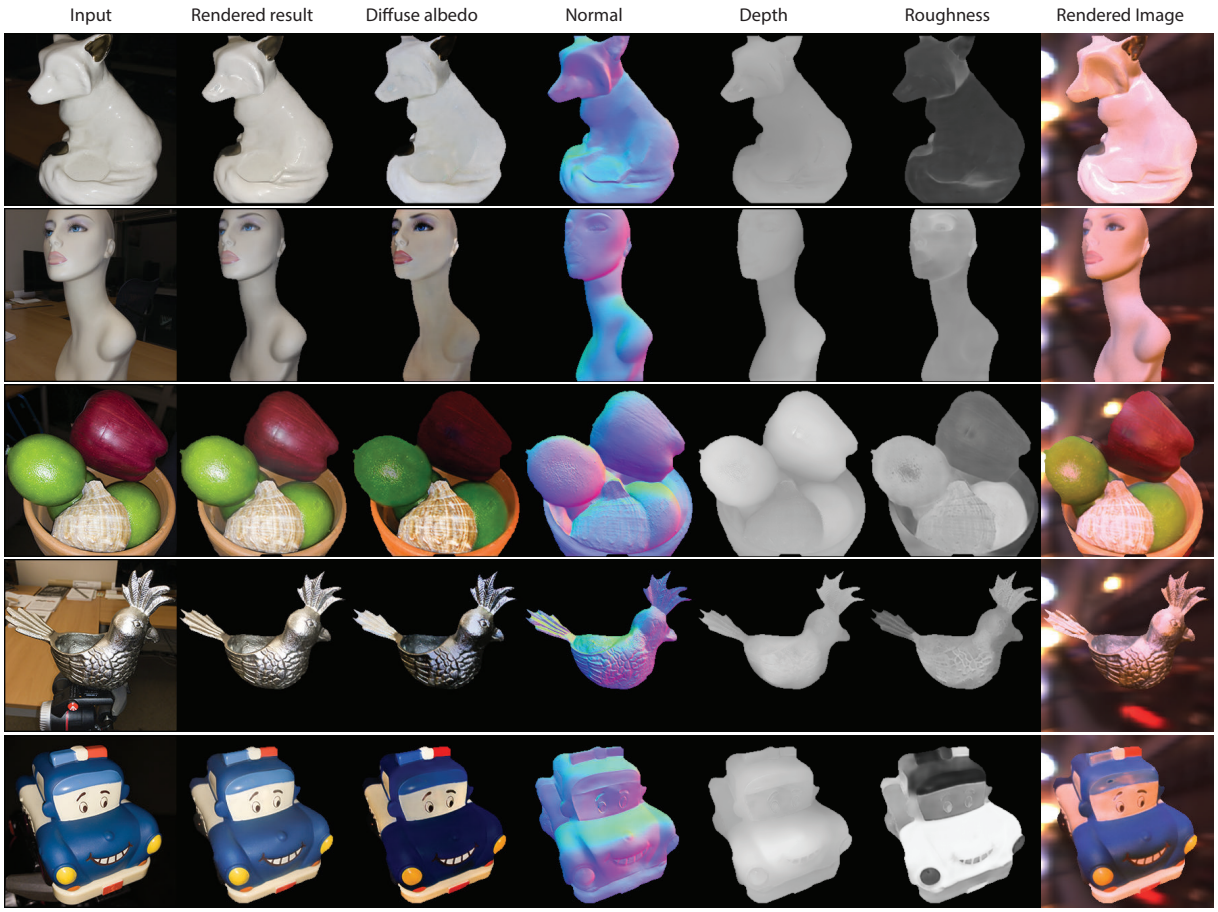


Figure 4.10. Results on real objects. For each example, we show the input image, the rendered output using the estimated shape and BRDF parameters, as well as visualization under a novel illumination condition. We observe high quality recovery of shape and spatially-varying BRDF.

important for it and the network may rely on connectivity priors to predict roughness. However, this prior may fail, which results in the same material having different roughness values (such as the owl in the second row of Figure 4.6). Such a prior might be explicitly enhanced to improve performance by using a densely connected CRF [171] or bilateral filter [14]. Another possibility would be to take shape-material correlations into account. From Figure 4.9, we can see that the error of depth prediction is significantly larger than the normal prediction, which suggests that we may use normal predictions to refine depth predictions [146]. Despite these limitations, we note that our network achieves significantly better results than prior works on this challenging, ill-posed problem.

4.5 Conclusion

In this chapter, we demonstrate the first approach for simultaneous estimation of arbitrary shape and spatially-varying BRDF, using a single mobile phone image. We make several physically-motivated and effective choices across image acquisition, dataset creation and network architecture. We use a mobile phone flash to acquire images, which allows observing high frequency details. Our large-scale dataset of procedurally created shapes, rendered with spatially-varying BRDF under various lighting conditions, prevents entanglement of category-level shape information with material properties. Our cascaded network allows global reasoning through error feedback and multiscale iterative refinement, to obtain highly accurate outputs for both shape and material. We propose a novel rendering layer to incorporate information from various lighting conditions, which must account for global illumination to handle arbitrary shape. Inspired by the physical process of rendering bounces of global illumination, we devise a cascaded CNN module that retains speed and simplicity. Extensive experiments validate our network design through high-quality estimation of shape and SVBRDF that outperforms previous methods.

In the rest of this thesis, we will explore in two directions for solving inverse rendering problems under more challenging setting. In the first direction, we will extend this framework to larger scenes, i.e. indoor scenes, which has more diverse materials, more complex spatially-varying lighting, shadows and global illumination (Chapter 6, 7 and 8). In the second direction, we will model more complex global illumination that cannot be handled by the learning-based rendering module proposed in this chapter – the refraction and reflection induced by transparent materials, which will be discussed in the next chapter.

Chapter 4 is based on the material as it appears in ACM Transactions on Graphics, 2018 (“Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image”, Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigator and author of this paper.

Chapter 5

Transparent Shape Reconstruction from Sparse Inputs

5.1 Introduction

Transparent objects abound in real-world environments, thus, their reconstruction from images has several applications such as 3D modeling and augmented reality. However, their visual appearance is far more complex than that of opaque objects as discussed in the prior two chapters, because of complex light paths with both refractions and reflections. This makes image-based reconstruction of transparent objects extremely ill-posed, since only highly convoluted intensities of an environment map are observed. In this chapter, we propose that data-driven priors learned by a deep network that models the physical basis of image formation can solve the problem of transparent shape reconstruction using a few natural images acquired with a commodity mobile phone camera.

While physically-based networks have been proposed in Chapter 4 to solve inverse problems for opaque objects, the complexity of light paths is higher for transparent shapes and small changes in shape can manifest as severely non-local changes in appearance. However, the physical basis of image formation for transparent objects is well-known – refraction at the interface is governed by Snell's law, the relative fraction of reflection is determined by Fresnel's equations and total internal reflection occurs when the angle of incidence at the interface to a medium with lower refractive index is below critical angle. These properties have been used to

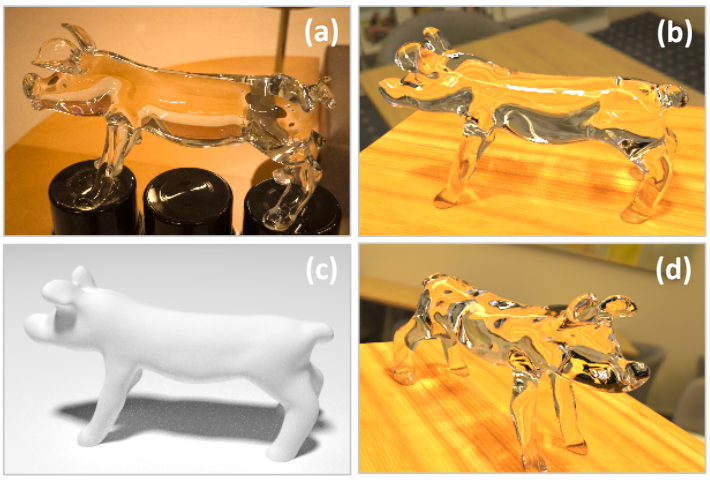


Figure 5.1. We present a novel physically-based deep network for image-based reconstruction of transparent objects with a small number of views. (a) An input photograph of a real transparent object captured under unconstrained conditions (1 of 10 images). (b) and (c): The reconstructed shape rendered under the same view with transparent and white diffuse material. (d) The reconstructed shape rendered under a novel view and environment map.

Figure 5.2. Reconstruction using 10 images of synthetic *kitten* model. The left image is rendered with the reconstructed shape while the right image is rendered with the ground-truth shape.

delineate theoretical conditions on reconstruction of transparent shapes [108], as well as acquire high-quality shapes under controlled settings [214, 231]. In contrast, we propose to leverage this knowledge of image formation within a deep network to reconstruct transparent shapes using relatively unconstrained images under arbitrary environment maps.

Specifically, we use a small number of views of a glass object with known refractive index, observed under a known but arbitrary environment map, using a mobile phone camera. Note that this is a significantly less restricted setting compared to most prior works that require dark room environments, projector-camera setups or controlled acquisition of a large number of images. Starting with a visual hull construction, we propose a novel in-network differentiable rendering layer that models refractive light paths up to two bounces to refine surface normals

corresponding to a backprojected ray at both the front and back of the object, along with a mask to identify regions where total internal reflection occurs. Next, we propose a novel cost volume to further leverage correspondence between the input image and environment map, but with special considerations since the two sets of normal maps span a four-dimensional space, which makes conventional cost volumes from multiview stereo intractable. Using our differentiable rendering layer, we perform a novel optimization in latent space to regularize our reconstructed normals to be consistent with the manifold of natural shapes. To reconstruct the full 3D shape, we use PointNet++ [163] with novel mechanisms to map normal features to a consistent 3D space, new loss functions for training and architectural changes that exploit surface normals for better recovery of 3D shape.

Since acquisition of transparent shapes is a laborious process, it is extremely difficult to obtain large-scale training data with ground truth [195]. Thus, we render a synthetic dataset, using a custom GPU-accelerated ray tracer. To avoid category-specific priors, we follow the strategy that has been shown successful in Chapter 4 by rendering images of random shapes under a wide variety of natural environment maps. On both synthetic and real data, the benefits of our physically-based network design are clearly observed. Indeed, we posit that such physical modeling eases the learning for a challenging problem and improves generalization to real images. Figures 5.1 and 5.2 show example outputs on real and synthetic data. All code and data will be publicly released.

To summarize, we propose the following contributions that solve the problem of transparent shape reconstruction with a limited number of unconstrained views:

- A physically-based network for surface normal reconstruction with a novel differentiable rendering layer and cost volume that imbibe insights from image formation.
- A physically-based 3D point cloud reconstruction that leverages the above surface normals and rendering layer.
- Strong experimental demonstration using a photorealistically rendered large-scale dataset for

training and a small number of mobile phone photographs for evaluation.

5.2 Related Work

Multiview stereo

Traditional approaches [180] and deep networks [227] for multiview stereo have achieved impressive results. A full review is out of our scope, but we note that they assume photoconsistency for opaque objects and cannot handle complex light paths of transparent shapes.

Theoretical studies

In seminal work, Kutulakos and Steger [108] characterize the extent to which shape may be recovered given the number of bounces in refractive (and specular) light paths. Chari and Sturm [37] further constrain the system of equations using radiometric cues. Other works study motion cues [20, 141] or parametric priors [203]. We derive inspiration from such works to incorporate physical properties of image formation, by accounting for refractions, reflections and total internal reflections in our network design.

Controlled acquisition

Special setups have been used in prior work, such as light field probes [211], polarimetry [50, 83, 138], transmission imaging [101], scatter-trace photography [140], time-of-flight imaging [199] or tomography [202]. An external liquid medium [73] or moving spotlights in video [229] have been used too. Wu et al. [214] also start from a visual hull like us, to estimate normals and depths from multiple views acquired using a turntable-based setup with two cameras that image projected stripe patterns in a controlled environment. A projector-camera setup is also used by [166]. In contrast to all of the above works, we only require unconstrained natural images, even obtainable with a mobile phone camera, to reconstruct transparent shapes.

Environment matting

Environment matting uses a projector-camera setup to capture a composable map [245, 46]. Subsequent works have extended to mutliple cameras [133], natural images [212], frequency

[244] or wavelet domains [158], with user-assistance [231], compressive sensing to reduce the number of images [56, 164] or deep network to predict the refractive flow from a single image [40]. In contrast, we use a small number of unconstrained images acquired with a mobile phone in arbitrary scenes, to produce full 3D shape.

Reconstruction from natural images

Stets et al. [194] propose a black-box network to reconstruct depth and normals from a single image. Shan et al. [183] recover height fields in controlled settings, while Yeung et al. [230] have user inputs to recover normals. In contrast, we recover high-quality full 3D shapes and normals using only a few images of transparent objects, by modeling the physical basis of image formation in a deep network.

Refractive materials besides glass

Polarization [44], differentiable rendering [38] and neural volumes [124] have been used for translucent objects, while specular objects have been considered under similar frameworks as transparent ones [84, 246]. Gas flows [7, 90], flames [85, 218] and fluids [71, 165, 237] have been recovered, often in controlled setups. Our experiments are focused on glass, but similar ideas might be applicable for other refractive media too.

5.3 Method

Setup and assumptions

Our inputs are N images $\{\mathbf{I}_n\}_{n=1}^N$ of a transparent object with known refractive index (IoR), along with segmentation masks $\{\mathbf{M}_n\}_{n=1}^N$. We assume a known and distant, but otherwise arbitrary, environment map $\mathbf{E}$. The output is a point cloud reconstruction $\{\mathbf{p}\}$ of the transparent shape. Note that our model is different from (3-2-2) triangulation [106] that requires two reference points on each ray for reconstruction, leading to a significant relaxation over prior works [214, 231] that need active lighting, carefully calibrated devices and controlled environments. We tackle this severely ill-posed problem through a novel physically-based network that models

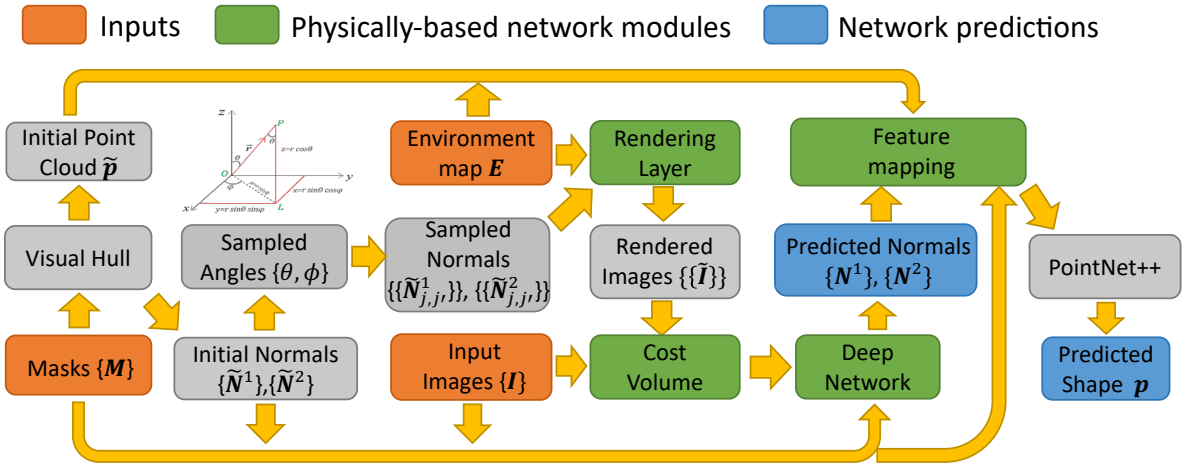


Figure 5.3. Our framework for transparent shape reconstruction.

the image formation in transparent objects over three sub-tasks: shape initialization, cost volume for normal estimation and shape reconstruction.

To simplify the problem and due to GPU memory limits, we consider light paths with only up to two bounces, that is, either the light ray gets reflected by the object once before hitting the environment map or it gets refracted by it twice before hitting the environment map. This is not a severe limitation – more complex regions stemming from total internal reflection or light paths with more than two bounces are masked out in one view, but potentially estimated in other views. The overall framework is summarized in Figure 5.3.

Shape initialization

We initialize the transparent shape with a visual hull [107]. While a visual hull method cannot reconstruct some concave or self-occluded regions, it suffices as initialization for our network. We build a 3D volume of size 128^3 and project segmentation masks from N views to it. Then we use marching cubes to reconstruct the hull and loop L3 subdivision to obtain smooth surfaces.

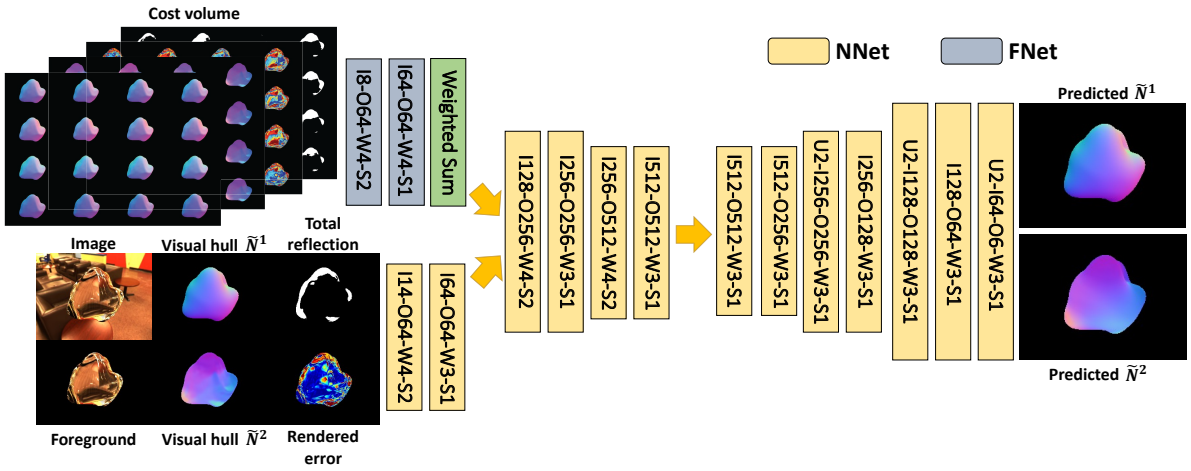


Figure 5.4. The network architecture for normal reconstruction. Yellow blocks represent **NNet** and blue blocks represent **FNet**. IX_1-OX_2-WX_3-SX_4 represents a convolutional layer with input channel X_1, output channel X_2, kernel size X_3 and stride X_4. UX_5 represents bilinear upsampling layer with scale factor X_5.

5.3.1 Normal Reconstruction

A visual hull reconstruction from limited views might be inaccurate, besides missed concavities. We propose to reconstruct high quality normals by estimating correspondences between the input image and the environment map. This is a very difficult problem, since different configurations of transparent shapes may lead to the same appearance. Moreover, small perturbations of normal directions can cause pixel intensities to be completely different. Thus, strong shape priors are necessary for a high quality reconstruction, which we propose to learn with a physically-motivated deep network.

Basic network

Our basic network architecture for normal estimation is shown in Figure 5.4. The basic network structure consists of one encoder and one decoder. The outputs of our network are two normal maps $\mathbf{N}^1$ and $\mathbf{N}^2$, which are the normals at the first and second hit points $\mathbf{p}^1$ and $\mathbf{p}^2$ for a ray backprojected from camera passing through the transparent shape, as illustrated in Figure 5.5(a). The benefit of modeling the estimation through $\mathbf{N}^1$ and $\mathbf{N}^2$ is that we can easily use a network to represent complex light transport effects without resorting to ray-tracing, which is

time-consuming and difficult to treat differentiably. In other words, given $\mathbf{N}^1$ and $\mathbf{N}^2$, we can directly compute outgoing ray directions after passage through the transparent object. The inputs to our network are the image $\mathbf{I}$, the image with background masked out $\mathbf{I} \odot \mathbf{M}$ and the $\tilde{\mathbf{N}}^1$ and $\tilde{\mathbf{N}}^2$ of the visual hull (computed off-line by ray tracing). We also compute $\hat{\mathbf{N}}^1$ and $\hat{\mathbf{N}}^2$ of the ground-truth shape for supervision. The definition of $\tilde{\mathbf{N}}^1$, $\tilde{\mathbf{N}}^2$ and $\hat{\mathbf{N}}^1$, $\hat{\mathbf{N}}^2$ are visualized in Figure 5.5(b). The basic network estimates:

$$\mathbf{N}^1, \mathbf{N}^2 = \mathbf{NNet}(\mathbf{I}, \mathbf{I} \odot \mathbf{M}, \tilde{\mathbf{N}}^1, \tilde{\mathbf{N}}^2) \tag{5.1}$$

The loss function is simply the L_2 loss for $\mathbf{N}^1$ and $\mathbf{N}^2$.

$$\mathscr{L}_n = ||\mathbf{N}^1 - \hat{\mathbf{N}}^1||_2^2 + ||\mathbf{N}^2 - \hat{\mathbf{N}}^2||_2^2 \tag{5.2}$$

Rendering layer

Given the environment map $\mathbf{E}$, we can easily compute the incoming radiance through direction $\mathbf{l}$ using bilinear sampling. This allows us to build a differentiable rendering layer to model the image formation process of refraction and reflection through simple local computation. As illustrated in Figure 5.5(a), for every pixel in the image, the incident ray direction $\mathbf{l}^i$ through that pixel can be obtained by camera calibration. The reflected and refracted rays $\mathbf{l}^r$ and $\mathbf{l}^t$ can be computed using $\mathbf{N}^1$ and $\mathbf{N}^2$, following Snell's law. Our rendering layer implements the full physics of an intersection, including the intensity changes caused by the Fresnel term F of the refractive material, which is defined in (2.7)

Due to total internal reflection, some rays entering the object may not be able to hit the environment map after one more bounce, for which our rendering layer returns a binary mask, $\mathbf{M}^{tr}$. With $\mathbf{I}^r$ and $\mathbf{I}^t$ representing radiance along the directions $\mathbf{l}^r$ and $\mathbf{l}^t$, the rendering layer models the image formation process for transparent shapes through reflection, refraction and

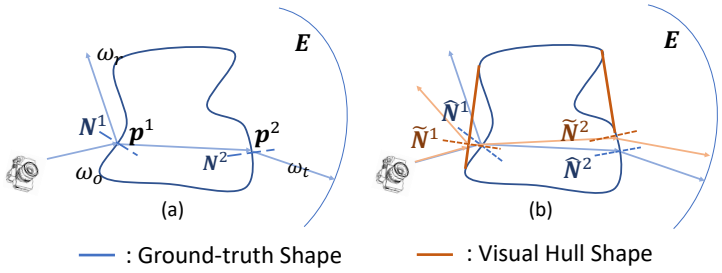


Figure 5.5. (a) Illustration of the first and second normal ($\mathbf{N}^1$ and $\mathbf{N}^2$), the first and second hit points ($\mathbf{p}^1$ and $\mathbf{p}^2$), and the reflection and refraction modeled by our deep network. (b) Illustration of visual hull ($\tilde{\mathbf{N}}^1$, $\tilde{\mathbf{N}}^2$) and ground-truth normals ($\hat{\mathbf{N}}^1$, $\hat{\mathbf{N}}^2$).

total internal reflection:

$$\mathbf{I}^r, \mathbf{I}^t, \mathbf{M}^{tr} = \textbf{RenderLayer}(\mathbf{E}, \mathbf{N}^1, \mathbf{N}^2). \tag{5.3}$$

Our in-network rendering layer is differentiable and end-to-end trainable. But instead of just using the rendering loss as an extra supervision, we compute an error map based on rendering with the visual hull normals:

$$\tilde{\mathbf{I}}^r, \tilde{\mathbf{I}}^t, \tilde{\mathbf{M}}^{tr} = \textbf{RenderLayer}(\mathbf{E}, \tilde{\mathbf{N}}^1, \tilde{\mathbf{N}}^2), \tag{5.4}$$

$$\tilde{\mathbf{I}}^{er} = |\mathbf{I} - (\tilde{\mathbf{I}}^r + \tilde{\mathbf{I}}^t)| \odot \mathbf{M}. \tag{5.5}$$

This error map is used as an additional input to our normal reconstruction network, to help it better learn regions where the visual hull normals $\tilde{N}^1$ and $\tilde{N}^2$ may not be accurate:

$$\mathbf{N}^1, \mathbf{N}^2 = \textbf{NNet}(\mathbf{I}, \mathbf{I} \odot \mathbf{M}, \tilde{\mathbf{N}}^1, \tilde{\mathbf{N}}^2, \tilde{\mathbf{I}}^{er}, \tilde{\mathbf{M}}^{tr}) \tag{5.6}$$

Cost volume

We now propose a cost volume to leverage the correspondence between the environment map and the input image. While cost volumes in deep networks have led to great success for multiview depth reconstruction of opaque objects, extension to normal reconstruction for transparent

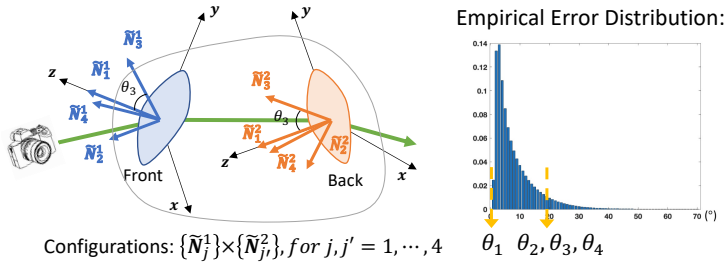


Figure 5.6. We build an efficient cost volume by sampling directions around visual hull normals according to their error distributions.

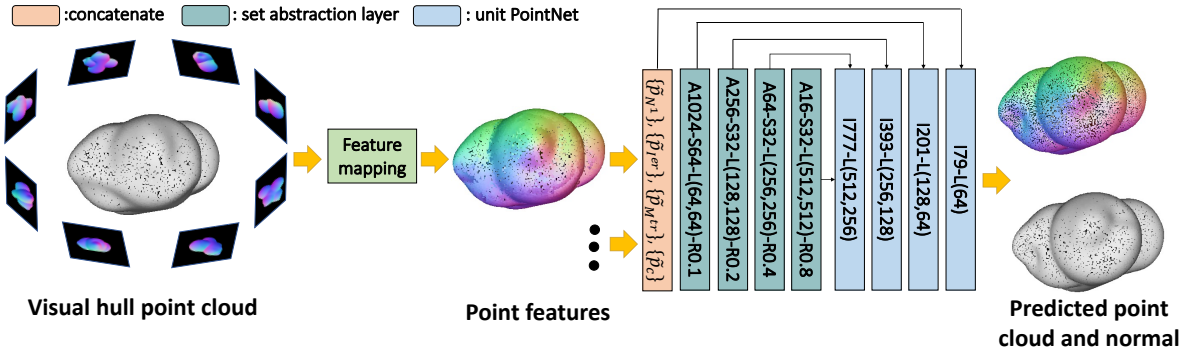
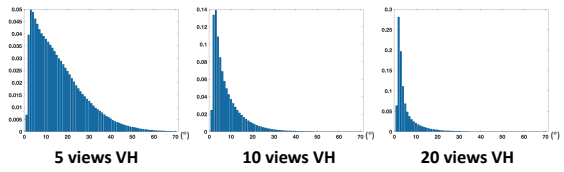


Figure 5.7. Our method for point cloud reconstruction. AX_1-SX_2-$L(X_3, X_4)$-RX_5 represents a set abstraction layer with X_1 anchor points, X_2 sampled points, 2 fully connected layers with X_3, X_4 feature channels and sampling radius X_5. IY_1-$L(Y_2, Y_3)$ represents a unit PointNet with Y_1 input channels and 2 fully connected layers with Y_2, Y_3 feature channels.

objects is non-trivial. The brute-force approach would be to uniformly sample the 4-dimensional hemisphere of $\mathbf{N}^1 \times \mathbf{N}^2$, then compute the error map for each sampled normal. However, this will lead to much higher GPU memory consumption compared to depth reconstruction due to higher dimensionality of the sampled space. To limit memory consumption, we sample $\mathbf{N}^1$ and $\mathbf{N}^2$ in smaller regions around the initial visual hull normals $\tilde{\mathbf{N}}^1$ and $\tilde{\mathbf{N}}^2$, as shown in Figure 5.6. Formally, let $\mathbf{u}$ be the up vector in bottom-to-top direction of the image plane. We first build a local coordinate system with respect to $\tilde{\mathbf{N}}^1$ and $\tilde{\mathbf{N}}^2$:

$$\mathbf{z} = \tilde{\mathbf{N}}^i,\ \mathbf{y} = \mathbf{u} - (\mathbf{u} \cdot \tilde{\mathbf{N}}^i)\tilde{\mathbf{N}}^i, \mathbf{x} = \texttt{cross}(\mathbf{y}, \mathbf{z}), \tag{5.7}$$

Table 5.1. Left: error distribution of visual normal. Right: the sampled angles for building cost volume. We set the sampled angles according to the normal error of visual hull reconstructed by different number of views.



	$\{\theta_j\}_{j=1}^4$	$\{\phi_j\}_{j=1}^4$
5 view	0°,25°,25°,25°	0°,0°,120°,240°
10 views	0°,15°,15°,15°	0°,0°,120°,240°
20 views	0°,10°,10°,10°	0°,0°,120°,240°

where $\mathbf{y}$ is normalized and $i = 1,2$. Let $\{\theta_j\}_{j=1}^J$, $\{\phi_j\}_{j=1}^J$ be the sampled angles. Then, the sampled normals are:

$$\tilde{\mathbf{N}}_j^i = \mathbf{x}\cos\phi_j \sin\theta_j + \mathbf{y}\sin\phi_j \sin\theta_j + \mathbf{z}\cos\theta_j. \tag{5.8}$$

We sample the angles $\{\theta_j\}_{j=1}^J$, $\{\phi_j\}_{j=1}^J$ according to the error distribution of visual hull normals. In particular, we first randomly sample 100 scenes from our synthetic dataset and compute the angles between visual hull normals and ground truth normals. We set one θ value to be 0 and the other to larger than 85% of angles between the visual hull normal $\tilde{\mathbf{N}}^1$ and ground truth normal $\hat{\mathbf{N}}^1$. The distribution of visual hull normal $\tilde{\mathbf{N}}^1$ error for 5, 10 and 20 views and sampled angles are presented in Table 5.1. Since we reconstruct $\mathbf{N}^1$ and $\mathbf{N}^2$ simultaneously, the total number of configurations of sampled normals is $J \times J$. Directly using the J^2 sampled normals to build a cost volume is too expensive, so we use a learnable pooling layer to aggregate the features from each sampled normal configuration in an early stage. For each pair of $\tilde{\mathbf{N}}_j^1$ and $\tilde{\mathbf{N}}_{j'}^2$, we compute their total reflection mask $\tilde{\mathbf{M}}_{j,j'}^{tr}$ and error map $\tilde{\mathbf{I}}_{j,j'}^{er}$ using (5.4) and (5.5), then perform a feature extraction:

$$\mathbf{F}(j,j') = \mathbf{FNet}(\tilde{\mathbf{N}}_j^1, \tilde{\mathbf{N}}_{j'}^2, \tilde{\mathbf{I}}_{j,j'}^{er}, \tilde{\mathbf{M}}_{j,j'}^{tr}). \tag{5.9}$$

We then compute the weighted sum of feature vectors $\mathbf{F}(j,j')$ and concatenate them with the feature extracted from the encoder of **NNet** for normal reconstruction:

$$\mathbf{F} = \sum_{j}^{J}\sum_{j'}^{J} k(j,j')\mathbf{F}(j,j'), \tag{5.10}$$

where $k(j,j')$ are positive coefficients with sum equal to 1, that are also learned during the training process. The detailed network structure is shown in Figure 5.4.

Post processing

The network above already yields reasonable normal reconstruction. It can be further improved by optimizing the latent vector from the encoder to minimize the rendering error using the predicted normal $\mathbf{N}^1$ and $\mathbf{N}^2$:

$$\mathscr{L}_n^{\text{Opt}} = ||(\mathbf{I} - (\mathbf{I}^r + \mathbf{I}^t)) \odot \mathbf{M}^{tr}||_2^2, \tag{5.11}$$

where $\mathbf{I}_t, \mathbf{I}_t, \mathbf{M}^{tr}$ are obtained from the rendering layer (5.3). For this optimization, we keep the network parameters unchanged and only update the latent vector. Note that directly optimizing the predicted normal $\mathbf{N}^1$ and $\mathbf{N}^2$ without the deep network does not yield comparable improvements. This is due to our decoder acting as a regularization that prevents the reconstructed normal from deviating from the manifold of natural shapes during the optimization. Similar ideas have been used for BRDF reconstruction [60].

5.3.2 Point Cloud Reconstruction

We now reconstruct the transparent shape based on the predictions of **NNet**, that is, the normals, total reflection mask and rendering error. Our idea is to map the predictions from different views to the visual hull geometry. These predictions are used as input features for a point cloud reconstruction to obtain a full 3D shape. Our point cloud reconstruction pipeline is illustrated in Figure 5.7.

Feature mapping

We propose three options to map predictions from different views to the visual hull geometry. Let $\{\tilde{\mathbf{p}}\}$ be the point cloud uniformly sampled from visual hull surfaces and $\mathbf{P}_n(\tilde{\mathbf{p}}, \mathbf{X})$ be a function that projects the 3D point $\tilde{\mathbf{p}}$ to the 2D image plane of view n and then fetches the value of feature $\mathbf{X}$ defined on image coordinates using bilinear sampling. Let $V_n(\tilde{\mathbf{p}})$ be a binary visibility function that verifies if point $\tilde{\mathbf{p}}$ can be observed from view n and $\mathbf{T}_n(\tilde{\mathbf{p}})$ be a transformation that maps a 3D point or normal direction in view n to world coordinates. Let $\cos_n(\tilde{\mathbf{p}})$ be the cosine of the angle between the ray passing through $\tilde{\mathbf{p}}$ and camera center.

The first option is a feature $\mathbf{f}$ that averages observations from different views. For every view n that can see the point $\tilde{\mathbf{p}}$, we project its features to the point and compute a mean:

$$\tilde{\mathbf{p}}_{\mathbf{N}^1} = \frac{\sum_n \mathbf{T}_n(\mathbf{P}_n(\tilde{\mathbf{p}}, \mathbf{N}_n^1))V_n(\tilde{\mathbf{p}})}{\sum_n V_n(\tilde{\mathbf{p}})}, \quad \tilde{\mathbf{p}}_{\mathbf{I}^{er}} = \frac{\sum_n \mathbf{P}_n(\tilde{\mathbf{p}}, \mathbf{I}_n^{er})V_n(\tilde{\mathbf{p}})}{\sum_n V_n(\tilde{\mathbf{p}})},$$
$$\tilde{\mathbf{p}}_{\mathbf{M}^{tr}} = \frac{\sum_n \mathbf{P}_n(\tilde{\mathbf{p}}, \mathbf{M}_n^{tr})V_n(\tilde{\mathbf{p}})}{\sum_n V_n(\tilde{\mathbf{p}})}, \quad \tilde{\mathbf{p}}_c = \frac{\sum_n \cos_n(\tilde{\mathbf{p}})V_n(\tilde{\mathbf{p}})}{\sum_n V_n(\tilde{\mathbf{p}})}.$$

We concatenate to get: $\mathbf{f} = [\tilde{\mathbf{p}}_{\mathbf{N}^1}, \tilde{\mathbf{p}}_{\mathbf{I}^{er}}, \tilde{\mathbf{p}}_{\mathbf{M}^{tr}}, \tilde{\mathbf{p}}_c]$.

Another option is to select a view n^* with potentially the most accurate predictions and compute f using the features from only that view. We consider two view-selection strategies. The first is nearest view selection, in which we simply select n^* with the largest $C_n(\tilde{\mathbf{p}})$. The other is to choose the view with the lowest rendering error and no total reflection. Note that although we do not directly map $\mathbf{N}^2$ to the visual hull geometry, it is necessary for computing the rendering error and thus, needed for our shape reconstruction.

Point cloud refinement

We build a network following PointNet++ [163] to reconstruct the point cloud of the transparent object. The input to the network is the visual hull point cloud $\{\tilde{\mathbf{p}}\}$ and the feature vectors $\{\mathbf{f}\}$. The outputs are the normals $\{\mathbf{N}_p\}$ and the offset of visual hull points $\{\delta\tilde{\mathbf{p}}\}$, with the

final vertex position is computed as $\mathbf{p} = \tilde{\mathbf{p}} + \delta\tilde{\mathbf{p}}$:

$$\{\delta\tilde{\mathbf{p}}\}, \{\mathbf{N}_p\} = \mathbf{PNet}(\{\tilde{\mathbf{p}}\}, \{\mathbf{f}\}). \tag{5.12}$$

Our lost function, $\mathscr{L}_p^{\mathrm{CD}}$, is based on the chamfer distance. Let $\{\hat{\mathbf{p}}\}$ be the set of points uniformly sampled from the ground-truth geometry, with normals $\hat{\mathbf{N}}_p$. Let $\mathbf{Ne}(\mathbf{p}, \{\hat{\mathbf{p}}\})$ be a function which finds the nearest point of $\mathbf{p}$ in the point set $\{\hat{\mathbf{p}}\}$ and function $\mathbf{Ne}_n(\mathbf{p}, \{\hat{\mathbf{p}}\})$ return the normal of the nearest point. The chamfer distance loss is defined as

$$\begin{aligned}\mathscr{L}_p^{\mathrm{CD}} = &\sum_{\{\mathbf{p}\},\{\mathbf{N}_p\}} \frac{k_1}{2}||\mathbf{p} - \mathbf{Ne}(\mathbf{p}, \{\hat{\mathbf{p}}\})|| + \frac{k_2}{2}||\mathbf{N} - \mathbf{Ne}_n(\mathbf{p}, \{\hat{\mathbf{p}}\})|| + \\ &\sum_{\{\hat{\mathbf{p}}\},\{\hat{\mathbf{N}}_p\}} \frac{k_1}{2}||\hat{\mathbf{p}} - \mathbf{Ne}(\hat{\mathbf{p}}, \{\mathbf{p}\})|| + \frac{k_2}{2}||\hat{\mathbf{N}}_p - \mathbf{Ne}_n(\hat{\mathbf{p}}, \{\mathbf{p}\})||.\end{aligned} \tag{5.13}$$

We set $k_1 = 200$ and $k_2 = 5$.

Our network, shown in Figure 5.7, makes several improvements over standard PointNet++. First, we replace max-pooling with average-pooling to favor smooth results. Second, we concatenate normals $\{\mathbf{N}\}$ to all skip connections to learn details. Third, we augment the input feature of set abstraction layer with the difference of normal directions between the current and center points. Section 5.4 show the impact of our design choices.

5.4 Experiments

Dataset

We procedurally generate random scenes following Chapter 4 rather than use shape repositories [36], to let the model be category-independent. To remove inner structures caused by shape intersections and prevent false refractions, we render 75 depth maps and use PSR [100] to fuse them into a mesh, with L3 loop subdivision to smooth the surface. We implement a physically-based GPU renderer using NVIDIA OptiX [157]. With 1499 HDR environment maps of [61] for training and 424 for testing, we render 3000 random scenes for training and 600 for

Table 5.2. Quantitative comparisons of normal estimation from 10 views. `vh10` represents the initial normals reconstructed from 10 views visual hull. `wr` and `basic` are our basic encoder-decoder network with and without rendering error map ($\mathbf{I}^{er}$) and total reflection mask ($\mathbf{M}^{tr}$) as inputs. `wr+cv` represents our network with cost volume. `wr+cv+op` represents the predictions after optimizing the latent vector to minimize the rendering error. `wr+cv var. IoR` represents sensitivity analysis for IoR, explained in text.

	vh10	basic	wr	wr+cv	wr+cv +op	wr+cv var. IoR
$\mathbf{N}^1$ median (°)	5.5	3.5	3.5	3.4	**3.4**	3.6
$\mathbf{N}^1$ mean (°)	7.5	4.9	5.0	4.8	**4.7**	5.0
$\mathbf{N}^2$ median (°)	9.2	6.9	6.8	6.6	**6.2**	7.3
$\mathbf{N}^2$ mean (°)	11.6	8.8	8.7	8.4	**8.1**	9.1
Render Err.(10^{-2})	6.0	4.7	4.6	4.4	**2.9**	5.5

testing. The IoR of all shapes is set to 1.4723, to match our real objects. Our experiments also include sensitivity analysis to characterize the behavior of the network when the test-time IoR differs from this value.

Implementation Details

When building the cost volume for normal reconstruction, we set the number of sampled angles J to be 4. Increasing the number of sampled angles will drastically increase the memory consumption and does not improve the normal accuracy. We sample ϕ uniformly from 0 to 2π and sample θ according to the visual hull normal error. We use Adam optimizer to train all our networks. The initial learning rate is set to be 10^{-4} and we halve the learning rate every 2 epochs. All networks are trained over 10 epochs.

5.4.1 Ablation Studies on Synthetic Data

Normal reconstruction

The quantitative comparisons of 10 views normal reconstruction are summarized in Table 5.2. We report 5 metrics: the median and mean angles of the first and the second normals ($\mathbf{N}^1$, $\mathbf{N}^2$), and the mean rendering error ($\mathbf{I}^{er}$). We first compare the normal reconstruction of the basic encoder-decoder structure with (`wr`) and without rendering error and total reflection mask as

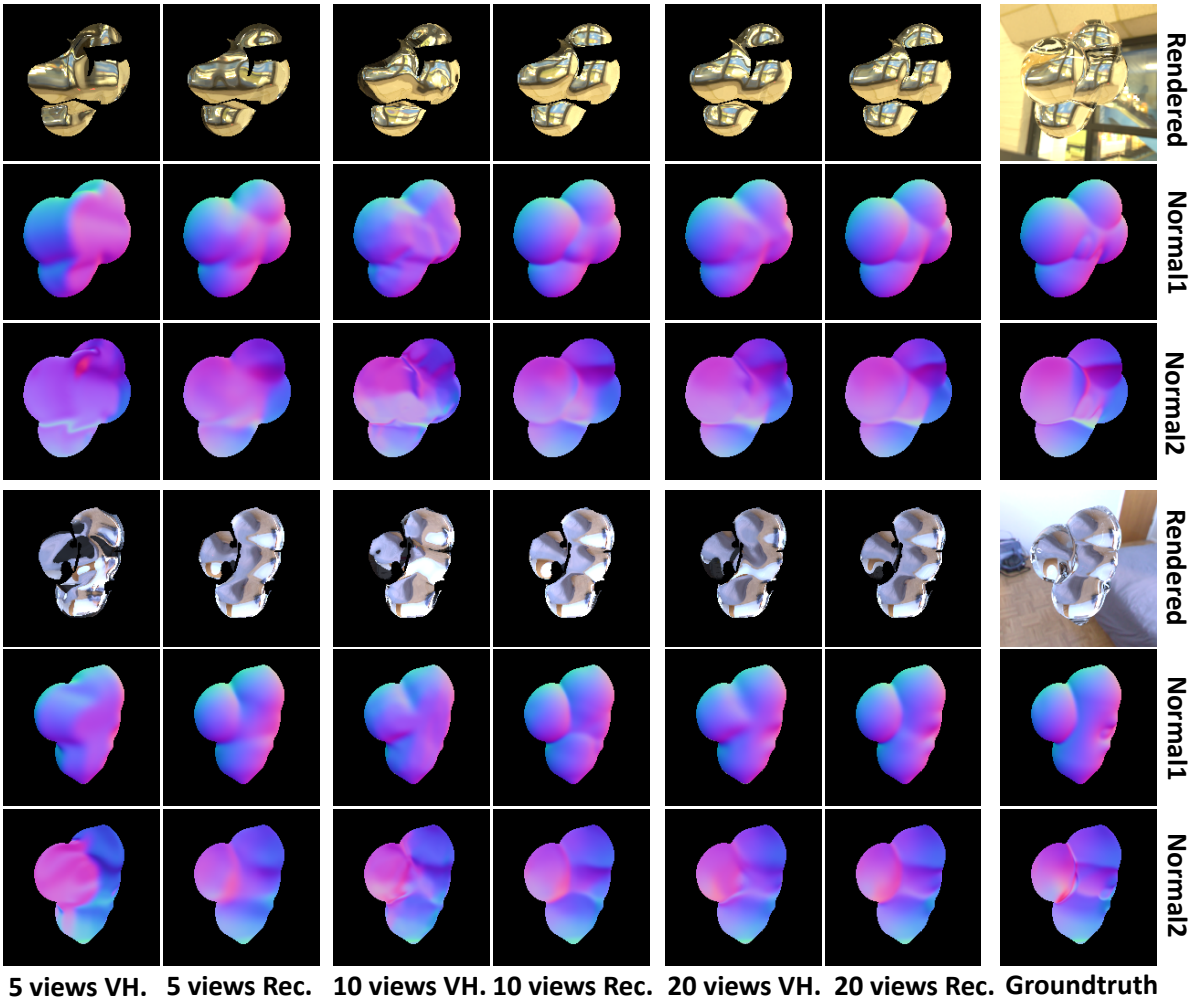


Figure 5.8. Two examples of normal reconstruction from our synthetic dataset with different input views. The region of total reflection has been masked out in the rendered images.

input (`basic`). While both networks greatly improve the normal accuracy compared to visual hull normals (`vh10`), adding rendering error and total reflection mask as inputs can help achieve overall slightly better performances. Next we test the effectiveness of the cost volume (`wr+cv`). Quantitative numbers show that adding cost volume achieves better results, which coincides with our intuition that finding the correspondences between input image and the environment map can help our normal prediction. Finally we optimize the latent vector from the encoder by minimizing the rendering error (`wr+cv+op`). It significantly reduces the rendering error and also improves the normal accuracy. Such improvements cannot be achieved by directly optimizing

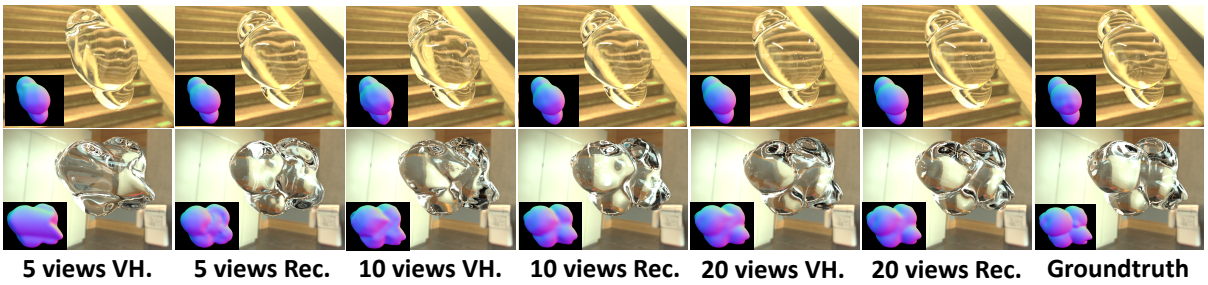


Figure 5.9. Our transparent shape reconstruction results from 5 views, 10 views and 20 views from our synthetic dataset. The images rendered with our reconstructed shapes are much closer to the ground-truth compared with images rendered with the visual hull shapes. The inset normals are rendered from the reconstructed shapes.

Table 5.3. Quantitative comparisons of point cloud reconstruction from 10 views. `RE`, `NE` and `AV` represent feature mapping methods: rendering error based view selection, nearest view selection and average fusion, respectively. `var. IoR` represents sensitivity analysis for IoR, as described in text. PSR represents optimization [100] to refine the point cloud based on predicted normals.

	CD(10^{-4})	CDN-mean($^\circ$)	CDN-med($^\circ$)	Metro(10^{-3})
`vh10`	5.14	7.19	4.90	15.2
`NE`	2.04	6.10	4.46	6.02
`AV`	2.03	6.08	4.46	6.09
`RE`	**2.00**	**6.02**	**4.38**	**5.98**
`RE, var. IoR`	2.13	6.24	4.56	6.11
PSR	5.13	6.94	4.75	14.7

the normal predictions $\mathbf{N}^1$ and $\mathbf{N}^2$ in the pixel space. Figure 5.8 presents normal reconstruction results from our synthetic dataset. Our normal reconstruction pipeline obtains results of much higher quality compared with visual hull method.

Point cloud reconstruction

Quantitative comparisons of the 10-view point cloud reconstruction network are summarized in Table 5.3. After obtaining the point and normal predictions $\{\mathbf{p}\}$ and $\{\mathbf{N}\}$, we reconstruct 3D meshes as described above. We compute the chamfer distance (CD), chamfer normal median angle (CDN-med), chamfer normal mean angle (CDN-mean) and Metro distance by uniformly sampling 20000 points on the ground-truth and reconstructed meshes. We test different feature

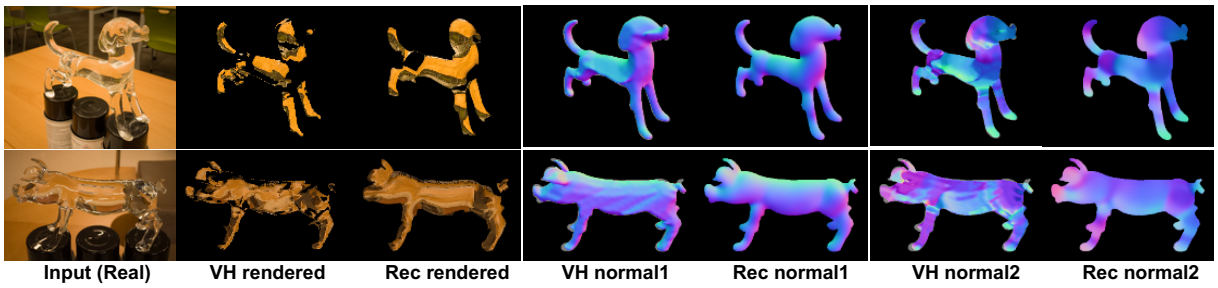


Figure 5.10. Normal reconstruction of real transparent objects and the rendered images. The initial visual hull normals are built from 10 views. The region of total reflection has been masked out in the rendered images.

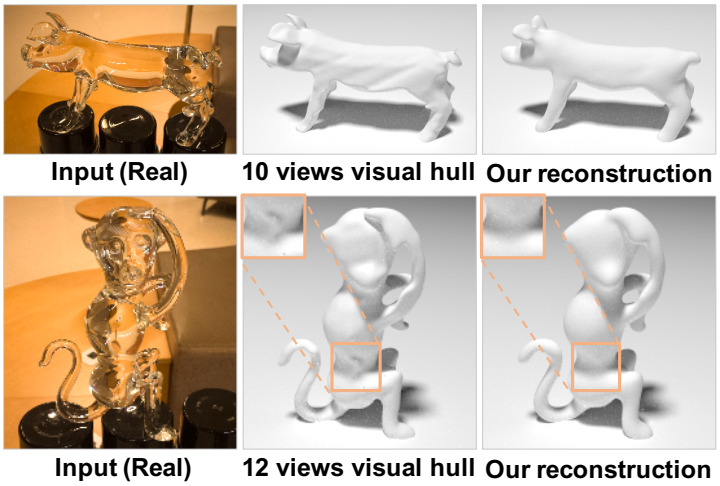


Figure 5.11. Comparison between visual hull initialization and our shape reconstruction on real objects. Our method recovers more details, especially for concave regions.

mapping strategies, where the rendering error based view selection method (RE) performs consistently better than the other two methods. This is because our rendering error can be used as a meaningful metric to predict normal reconstruction accuracy, which leads to better point cloud reconstruction.

The last row of Table 5.3 shows that an optimization-based method like PSR [100] to refine shape from predicted normals does not lead to much improvement, possibly since visual hull shapes are still significantly far from ground truth. In contrast, our network allows large improvements.

Table 5.4. Quantitative comparisons of point cloud reconstruction from 5 views and 20 views. In both cases, our pipeline significantly improves the transparent shape reconstruction accuracy compared with classical visual hull method.

	CD(10^{-4})	CDN-mean(°)	CDN-med(°)	Metro(10^{-3})
vh5	31.7	13.1	10.3	66.6
Rec5	**6.30**	**11.0**	**8.7**	**15.2**
vh20	2.23	4.59	**2.71**	6.83
Rec20	**1.20**	**4.04**	2.73	**4.18**

Table 5.5. Quantitative comparisons of transparent shape reconstruction on real data. We observe that our reconstruction achieves lower average errors than the visual hull method on all the metrics.

	Views	CD(10^{-4})		CDN-mean(°)		CDN-med(°)		Metro(10^{-3})	
		vh	Rec	vh	Rec	vh	Rec	vh	Rec
monkey	12	3.99	**3.94**	21.2	**16.4**	14.8	**11.9**	20.7	**13.9**
mouse	10	8.04	**5.35**	19.0	**16.3**	11.4	**12.0**	16.6	**13.0**
pig	10	5.58	**4.87**	19.0	**18.3**	**14.0**	14.6	13.0	**7.4**
dog	10	2.25	**1.86**	14.5	**12.4**	11.4	**10.3**	4.1	**4.0**
mean	10.5	4.97	**4.00**	18.4	**15.9**	12.9	**12.2**	13.6	**9.6**

Different number of views

We also test the entire reconstruction pipeline for 5 and 20 views, with results summarized in Table 5.4. We use the setting that leads to the best performance for 10 views, that is, $\texttt{wr} + \texttt{cv} + \texttt{op}$ for normal reconstruction and RE for view selection, achieving significantly lower errors than the visual hull method. Figure 5.8 and 5.9 shows examples from the synthetic test set for normal and point cloud reconstructions with different number of views.

Sensitivity analysis for IoR

We also evaluate the model on another test set with the same geometries, but unknown IoRs sampled uniformly from the range $[1.3, 1.7]$. As shown in Tables 5.2 and 5.3, errors increase slightly but stay reasonable, showing that our model can tolerate inaccurate IoRs to some extent.

5.4.2 Results on Real Transparent Objects

We acquire RGB images using a mobile phone. To capture the environment map, we take several images of a mirror sphere at the same location as the transparent shape. We use COLMAP [179] to obtain the camera poses and manually create the segmentation masks.

Normal reconstruction

We first demonstrate the normal reconstruction results on real transparent objects in Figure 5.10. Our model significantly improves visual hull normal quality. The images rendered from our predicted normals are much more similar to the input RGB images compared to those rendered from visual hull normals.

3D shape reconstruction

In Figure 5.12, we demonstrate our 3D shape reconstruction results on real world transparent objects under natural environment map. The companion video [4] also shows renderings of the reconstructed shapes under new lighting with different materials. We first demonstrate the reconstructed shape from the same view as the input images by rendering them under different lighting and materials. Even with very limited inputs, our reconstructed shapes are still of high quality. To test the generalizability of our predicted shapes, we render them from novel views that have not been used as inputs and the results are still reasonable. Figure 5.11 compares our reconstruction results with the visual hull initialization. We observe that our method performs much better, especially for concave regions. The quantitative numbers are summarized in Table 5.5. We manually align ground-truth shapes with the predicted shapes using ICP method [21] and then uniformly sample 20000 points on the both shapes to compute the four error metrics (CD, CDN-mean, CDN-med, Metro). For all the 4 objects, our method consistently outperforms the visual hull baseline, which again demonstrates the effectiveness of our transparent shape reconstruction framework.

Runtime

Our method requires around 46s to reconstruct a transparent shape from 10 views on a 2080 Ti, compared to 5-6 hours for previous optimization-based methods [214].

Limitations

Our limitations suggest interesting future avenues of research. A learnable multiview fusion might replace the visual hull initialization. We believe more complex light paths of length greater than 3 may be handled by differentiable path tracing along the lines of differentiable rendering [110, 234]. While we assume a known refractive index, it may be jointly regressed. Finally, since we reconstruct $\mathbf{N}^2$, future works may also estimate the back surface to achieve single-view 3D reconstruction.

5.5 Conclusion

We present the first physically-motivated deep network to reconstruct transparent shapes from a small number of views captured under arbitrary environment maps. Our network models the properties of refractions and reflections through a physically-based rendering layer and cost volume, to estimate surface normals at both the front and back of the object, which are used to guide a point cloud reconstruction. Extensive experiments on real and synthetic data demonstrate that our method can recover high quality 3D shapes.

Chapter 5 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020 ("Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes", Zhengqin Li*, Yu-ying Yeh*, Manmohan Chandraker). The dissertation author was the one of the primary investigators and author of this paper. Yu-ying Yeh is a co-first author who contributed equally to the paper.

Figure 5.12. Results on 3D reconstruction for four real transparent objects. All shapes are reconstructed from 10 views, except the *monkey* in the last row that uses 12 views. The odd rows show the input image and the reconstructed shapes under different lighting and materials while the corresponding outputs using the ground-truth are shown in the even rows. We also render the reconstructed shapes and ground-truth shapes from a novel view direction that has not been used to build the visual hull (columns 7-8).

Chapter 6

Indoor Scene Photorealistic Synthetic Dataset Creation

6.1 Introduction

In the next three chapters, we will address a long-standing challenge for both computer visiion and graphics, to *reconstruct geometry, spatially-varying complex reflectance and spatially-varying lighting and physically-based light sources* from a single RGB image of an arbitrary indoor scene captured under uncontrolled conditions, while supporting various photorealistic scene editing applications, ranging from virtual object insertion to light source editing. This is an extremely challenging problem – indoor scenes display the entire range of real-world appearance, including arbitrary geometry and layouts, localized light sources that lead to complex spatially-varying lighting effects and complex, non-Lambertian surface reflectance. Following our success in modeling complex scene appearance from Chapter 3 to 5, we propose a physically-motivated deep learning frameworks to regress these scene parameters holistically from an input image.

However, reasoning about these underlying, entangled factors requires large-scale high-quality ground truth, which remains hard to acquire. While ground truth geometry can be captured using a 3D scanner, it is extremely challenging (if not nearly impossible) to accurately acquire the complex spatially-varying materials and lighting of indoor scenes. An alternative is to consider synthetic datasets, but large-scale synthetic datasets of indoor scenes with plausible geometry, materials and lighting are also non-trivial to create.

Therefore, in this chapter, we present OpenRooms, an open framework for synthesizing photorealistic indoor scenes, with broad applicability across computer vision, graphics and robotics. It has several advantages over prior works, summarized in Table 6.1.

First, rather than use artist-created scenes and assets, we ascribe high-quality material and lighting to RGBD scans of real indoor scenes. Beyond just the data, we provide all the tools necessary to accomplish this, allowing any researcher to inexpensively create such datasets. While prior works can align CAD models to scanned point clouds [8, 88, 9], they do not explore how to assign materials and lighting appropriately to build a large-scale photorealistic dataset.

Second, we provide extensive high-quality ground truth for complex light transport that is unmatched in prior works. Our material is represented by a spatially-varying microfacet bidirectional reflectance distribution function (SVBRDF), and our lighting includes windows, environment maps and area lights, along with their per-pixel spatially-varying effects to account for visibility, shadows and inter-reflections. As will be shown in Chapter 7 and 8, these high-quality and extensive ground truth help better understand complex light transport in indoor scenes and enable new applications in photorealistic augmented reality, where we demonstrate virtual object insertion, material editing, light editing.

Third, we render photorealistic images with our data and tools, which include a custom GPU-accelerated physically-based renderer. We create an instance of such a dataset by building on existing repositories: 3D scans from ScanNet [51], CAD model alignment [8], reflectance [3] and illumination [74, 77]. The resulting dataset contains over 100K HDR photorealistic images, with various light transport effects being modeled accurately. Our dataset is publicly available and can be significantly extended through future community efforts based on our tools. We also demonstrate applicability of our method to other choices for material [2] and geometry [190].

In addition to inverse rendering, we believe that our effort will significantly accelerate research in multiple areas. We demonstrate that OpenRooms may be used for training semantic segmentation networks [239, 42], as well as multi-task learning to jointly estimate shape, material and semantics. Studies in robotics may also benefit by using our ground truth to enhance

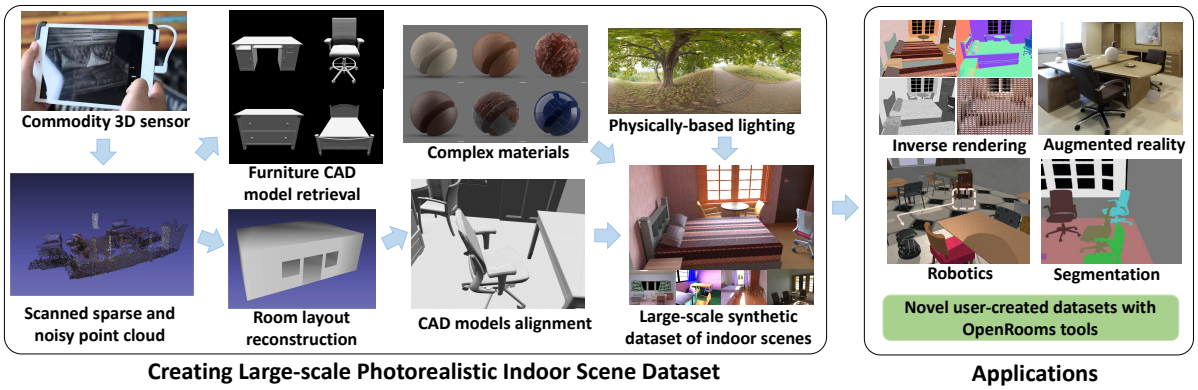


Figure 6.1. Our framework for creating a synthetic dataset of complex indoor scenes with ground truth shape, SVBRDF and SV-lighting, along with the resulting applications. Given possibly noisy scans acquired with a commodity 3D sensor, we generate consistent layouts for room and furniture. We ascribe per-pixel ground truth for material in the form of high-quality SVBRDF and for lighting as spatially-varying physically-based representations. We render a large-scale dataset of images associated with this ground truth, which can be used to train deep networks for inverse rendering and semantic segmentation. We further motivate applications for augmented reality and robotics, while suggesting that the open source tools we make available can be used by the community to create other large-scale datasets too. The accompanying video [4] demonstrates our high-quality rendered images, ground truths and the overall pipeline for photorealistic synthetic indoor dataset creation.

existing simulation environments [216, 178, 219, 128]. We demonstrate this possibility by combining OpenRooms assets with the PyBullet engine [1] and mapping our SVBRDFs to friction coefficients, to motivate navigation and rearrangement under different material and lighting. We also note that OpenRooms allows a one-to-one correspondence between real videos and simulations, which can be valuable for sim-to-real transfer [92].

In this chapter, we will illustrate the OpenRooms framework for creating large-scale, high-quality synthetic indoor datasets from commodity RGBD sensor scans, as shown in Figure 6.1. We will also demonstrate how it can enable scene understanding and robotics research but will present inverse rendering tasks, which is the focus of our research, in Chapter 7 and 8 in detail.

Table 6.1. OpenRooms is distinct in providing extensive ground truth for photorealism (especially material and lighting), with publicly available assets and tools. The tools in OpenRooms framework allow generating synthetic counterparts of real scenes, with high-quality ground truth.

Dataset	Available annotations						Publicly available assets				Corresponding real images and scenes
	Geometry	Material	Lighting			Segmentation	Images	CAD	Baseline	Tool	
			Light sources	Per-pixel	Visibility						
PBRS [238]	✓	diffuse	✗	shading	✗	✓	✗	✗	✓	✓	✗
Scenenet [134]	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✗
CGIntrinsic [114]	✗	diffuse	✗	shading	✗	✗	✓	✗	✓	✓	✗
InteriorNet [111]	✓	diffuse	✗	shading	✗	✓	✓	✗	✓	✓	✗
CG-PBR [181]	✓	phong	✗	shading	✗	✗	✗	✗	✗	✓	✗
InvIndoor [115]	✓	microfacet	✗	envmap	✗	✗	✗	✗	✓	✓	✗
3D-Future [59]	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗
AI2-THOR [105]	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✓
Structure3D [241]	✓	✗	✗	shading	✗	✓	✓	✗	✗	✓	✗
Hypersim [173]	✓	diffuse	✗	highlight	✗	✓	✓	✗	✗	✓	✗
OpenRooms	✓	microfacet	✓	envmap	✓	✓	✓	✓	✓	✓	✓

6.2 Related Work

The importance of indoor scene reconstruction and understanding has led to a number of real datasets [188, 51, 35, 219, 185]. While they are by nature photorealistic, they only capture some scene information (usually images, geometry and semantic labels). However, we are interested in studying geometry, reflectance and illumination, where the latter two are particularly challenging to acquire in real datasets. Synthetic datasets provide an alternative [134, 191, 111], but prior ones are limited with respect to rendering arbitrary data [111], scene layout [134], material [191], or baselines [173], as summarized in Table 6.1.

Several methods build 3D models for indoor scenes from a single image [88] or scans [8, 9, 29, 41]. However, our focus is beyond geometry, to assign real-world materials and lighting to create photorealistic scenes. To the best of our knowledge, the only existing dataset with complex materials and spatially-varying lighting annotations is from Li et al. [115], but is built on artist-created assets that are not publicly available [191]. We instead create photorealistic indoor scene datasets that start with 3D scans to provide high-quality ground truth for geometry, reflectance and lighting.

Several indoor virtual environments have also been proposed for robotics and embodied

Figure 6.2. Images from ScanNet and our corresponding synthetic scene layouts rendered with different materials, different lighting, and different views selected by our algorithm. The third row shows the same scene as the second one, but rendered with freely available Substance Share materials [2] instead of the public but non-free Adobe Stock materials [3].

vision [216, 178, 219, 185, 128, 105]. Our work is complementary, where our photorealistic ground truth and suite of tools could be used to enhance existing virtual environments and conduct new types of studies. In Section 6.4, we seek to motivate such adoption by illustrating integration with a physics engine and computing ground truth for friction coefficients.

6.3 Building a Photorealistic Indoor Dataset

We now describe our framework for building a synthetic dataset of complex indoor scenes. We demonstrate this using ScanNet, a large-scale repository of real indoor scans [51], but our work is also applicable to other datasets [190, 80]. We briefly describe the geometry creation, while focusing on our principal novelties of photorealistic material and lighting.

6.3.1 Creating CAD Models from 3D Scans

While recent methods such as [9] are possible alternatives, we demonstrate our dataset creation example utilizing existing labels in ScanNet and initial CAD alignment [8] to create the ground truth geometry robustly.

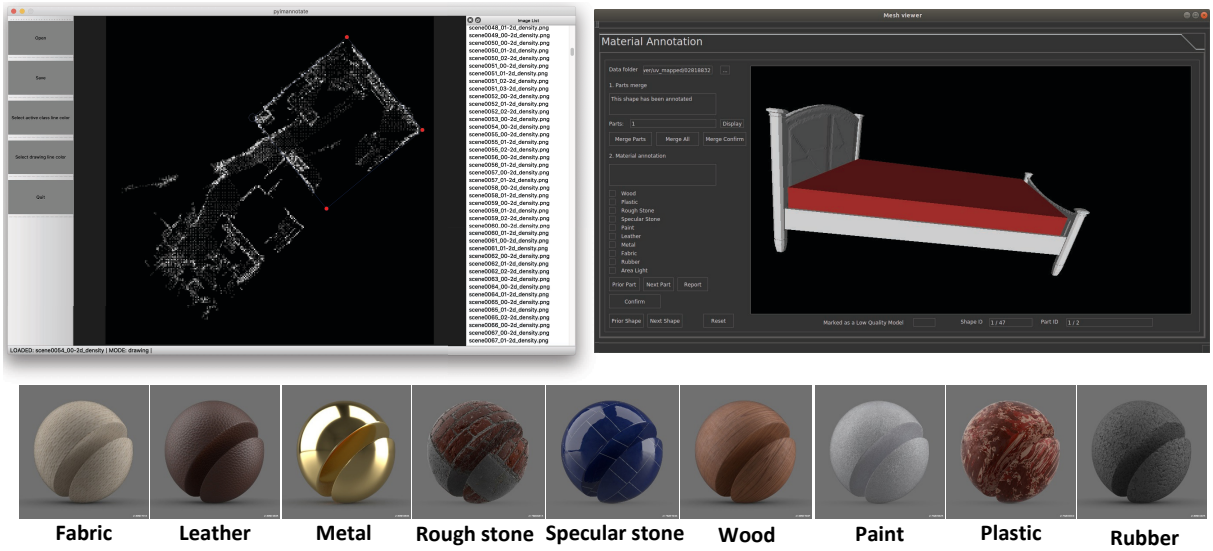


Figure 6.3. UIs for annotating room layout (Left top) and material category (Right top). (Bottom) Material examples from each category. Please zoom in for better visualization.

Reconstructing the room layout

We fuse the depth maps from different views of a scene to obtain a single point cloud. We design a UI for fast layout annotation (Figure 6.3), which projects the 3D point cloud to the floor plane and a polygon may be selected for the layout. While the annotation needs less than a minute per scene, we also train a Floor-SP network [41] on these annotations that users may employ for their own scenes. Next we use RANSAC to determine the horizontal floor plane. Since ScanNet views generally do not cover the ceiling, we assign a constant room height of 3 meters.

Windows and doors

Special consideration is needed for doors and windows as they are important illuminants in indoor scenes. We project the 3D points labeled as doors and windows to the closest wall, then divide the wall into segments and merge connected segments with sufficient number of points, to which a ShapeNet CAD model is assigned.

Figure 6.4. The first row shows images rendered with materials from our dataset. The second and third rows are images rendered with the original materials from SUNCG dataset using Lambertian and Phong models. Images rendered with our materials have realistic specular highlights.

Consistent furniture placement

We use initial poses from Scan2CAD [8] to align CAD models with furniture instances. We do not require appearances to closely match the input images, but generate plausible layouts and shapes with as much automation as possible. Our tool automatically moves bounding boxes for furniture perpendicular to the floors and walls to resolve floating objects and intersections. Such geometric consistency is important since our dataset may also be used for tasks such as navigation.

Semantic labels

Given our geometry ground truth, it is straightforward to obtain labels for semantic and instance segmentation based on PartNet annotations, as shown in Figure 6.5. We demonstate in experiments that our labels can be used to train single and multi-task deep networks.

6.3.2 Assigning Complex Materials to Indoor Scenes

One of the major contributions of our dataset is ground-truth annotation of complex material parameters for indoor scenes. Previous works typically provide material annotations as

simple diffuse or Phong reflectance [191, 186], while we provide a physically-based microfacet SVBRDF. As shown in

Assigning materials to ShapeNet

Many ShapeNet CAD models do not have texture coordinates, so we use Blender's [48] cube projection UV mapping to compute texture coordinates for them automatically. Inspired by Photoshape [156], we split CAD models into semantically meaningful parts and assign a material to each part. While Photoshape does this for only chairs, we do so for all furniture types in indoor scenes, using the semantically meaningful part segmentation of 24 categories of models provided by PartNet [139].

Material annotation UI

We design a custom UI tool to annotate material category for each part, as shown in Figure 6.3. It allows merging over-segmented parts which should be assigned the same material. To allow material annotation, we group 1,078 SVBRDFs into 9 categories based on their appearances, similar to Chapter 3, as shown in Figure 6.3. Annotators label a material category for each part, with a specific material sampled randomly from the category. While we do not pursue mimicking input appearances, we do seek that photorealism and semantics be respected in the dataset. Experiments show that our dataset created following the above choices enables state-of-the-art inverse rendering performances. Note our distinction from domain randomization, since arbitrary choices for material and lighting might not allow generalization on real scenes for extremely ill-posed problems like material and lighting estimation. Our tools and the annotations will be released for future research.

6.3.3 Ground Truth Lighting for Indoor Scenes

Lighting plays one of the most important roles in image formation. However, prior datasets usually only provide diffuse shading as their lighting representation [114, 238]. Recent work provides per-pixel environment maps by rendering the incoming radiance at every surface

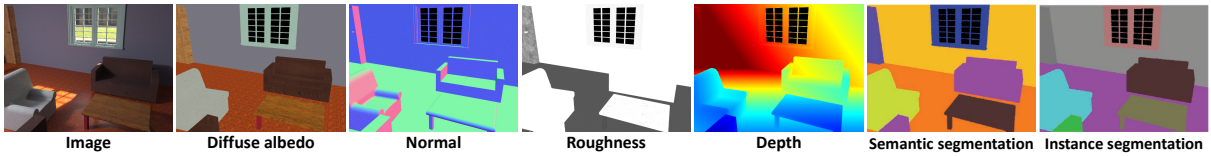


Figure 6.5. One of our rendered images with ground-truth geometry, spatially-varying material and segmentation labels.

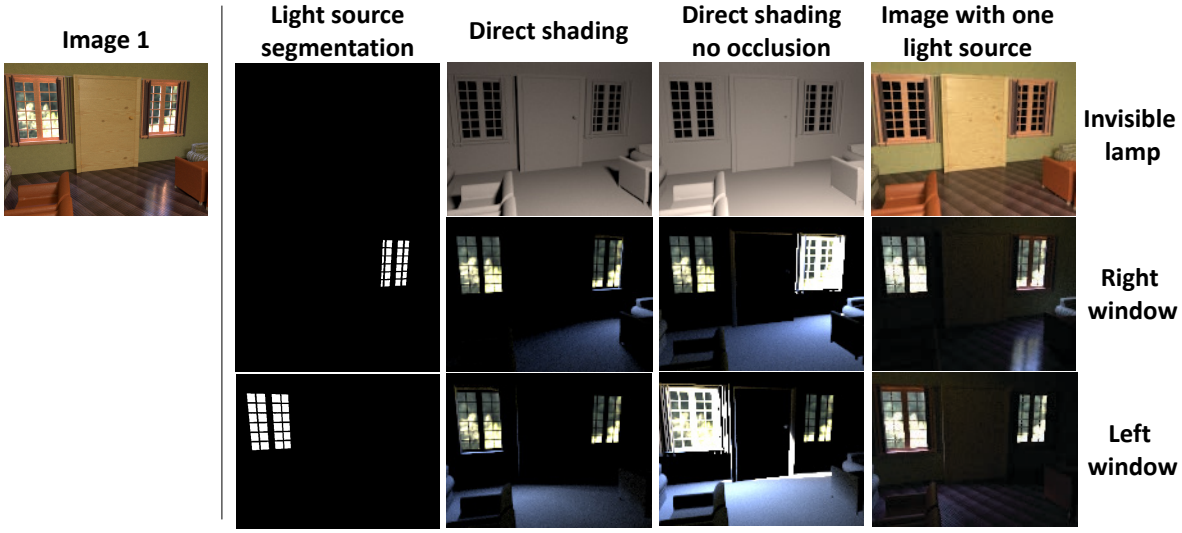


Figure 6.6. Our ground-truth light source annotations. From left to right: input and for each light source, its instance segmentation, and direct shading with and without occlusion. Our annotations reveal rich information about light transport in indoor scenes.

point in the camera frustum [115], which allows modeling shadows and specular highlights, but not the complex interactions among global light sources, scene geometry, materials and local lighting. On the contrary, OpenRooms provides extra supervision for visible and invisible light sources, the contribution of each individual light source to the local lighting, direct and indirect lighting, as well as visibility. Such rich supervision may help better understand the complex light transport in indoor scenes and enable new applications such as editing of light sources and dynamic scenes.

Light sources

We model two types of light sources in OpenRooms—windows and lamps—and we provide ground-truth annotations for them. The annotations include instance segmentation

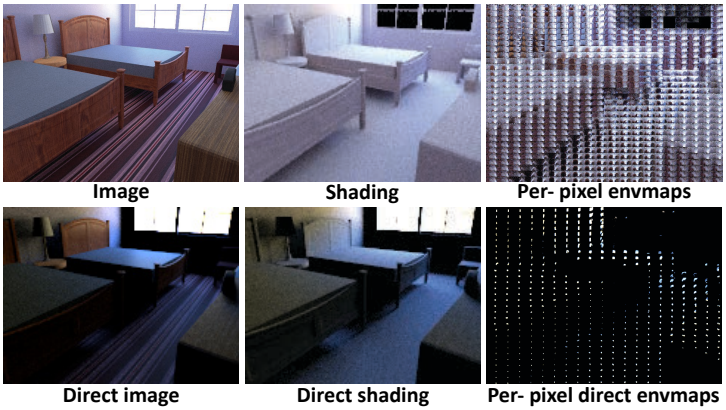


Figure 6.7. We provide various types of supervision for lighting analysis of indoor scenes, including per-pixel environment maps with only direct illumination, or including indirect illumination.

masks for visible light sources and a consistent parameterized representation for both visible and invisible light sources. More specifically, for each window, we model its geometry using a rectangular plane and the lighting coming through the window using an environment map rendered at its center. We represent each lamp as a 3D bounding box following the standard area light model. We visualize our light source annotations in Figure 6.6. Our light source representation has clear physical meaning and can model the full physics of image formation in indoor scenes.

Light source colors

For environment maps, we use 414 high-resolution HDR panoramas of natural outdoor scenes, from [77] and [74]. For indoor lamps, unlike previous synthetic datasets that randomly sample the spectrum of area lights [115, 238, 114], we follow a physically plausible black-body model to determine the spectrum of the light source by its temperature, chosen between 4000K to 8000K.

Per-pixel lighting

To enable tasks such as object insertion or material editing, we must estimate lighting at every spatial. We render per-pixel environment maps and shading as a spatially-varying lighting

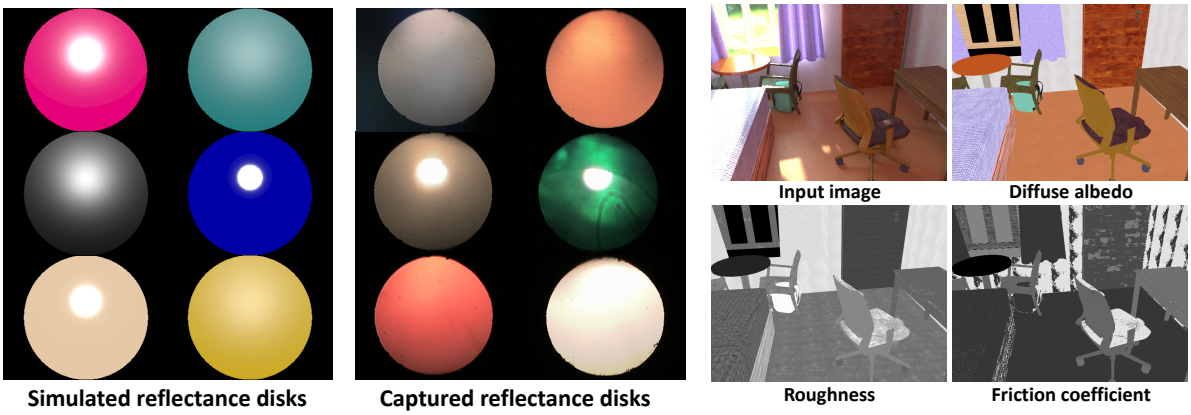


Figure 6.8. Left: Comparisons of randomly sampled reflectance disks captured by the system of Zhang et al. [236] and rendered by our virtual environment. Right: Visualization of friction coefficient in OpenRooms dataset. We map diffuse albedo and roughness parameters to friction coefficient based on nearest neighbor search. We observe that specular materials usually have smaller friction coefficients.

representation. For per-pixel environment map specifically, we render a 16×32 environment map at the corresponding 3D point on object surfaces at every pixel. We render both with direct, as well as combined direct and indirect illumination. This will help to separately analyze the direct contribution from light sources and indirect reflections from the indoor scene. We visualize an example in Figure 6.7.

Per-light direct shading and visibility

In order to understand complex light transport in indoor scenes, we also provide the separate contribution of every individual light source and its visibility map. For each image, we render the direct shading of each light source, with and without considering the occlusion term, by turning on only that particular light source. The visibility map can be computed as the ratio of the two direct shading images. We visualize these annotations in Figures 6.6. These will allow new challenging light editing tasks not possible with prior datasets, such as turning on and off a light or opening a window, which we will be discussed in Chapter 8.

6.3.4 Ground Truth Friction Coefficient

We follow the concept of reflectance disks from Zhang et al. [236] for predicting friction coefficients for various materials. The acquisition setup of [236] includes a beam splitter, an orthographic camera and a parabolic mirror, to capture material appearances by densely sampling from a large range of view directions and a small range of lighting directions (please see Figure 3 of [236]). We mimic this capture system to render the reflectance disk using our physically-based renderer. We uniformly sample the parameter space of our microfacet BRDF model and render a reflectance disk for each sampled point. Figure 6.8 left compares the reflectance disks rendered under our virtual environment and captured by the system. We observe that the distribution of specular highlights and intensities of the two sets of reflectance disks can match well.

After obtaining the reflectance disk, Zhang et al. [236] use a pretrained deep network to map the reflectance disk to a low dimensional latent space, which is termed a deep reflectance code. Due to the dense down-sampling operations, the deep reflectance code is robust to translation and rotation, which makes it a suitable representation for modeling intrinsic properties of materials, including the friction properties. Thereafter, they use K-nearest neighbor method to map deep reflectance code to friction coefficients. Following their implementation, we also map our reflectance disks to a deep reflectance code, to the friction coefficients using nearest neighbor search for each of our sampled microfacet BRDF parameters. This gives us a table that allows us to map our microfacet BRDF parameters to friction coefficients through bilinear interpolation or nearest neighbor search. Figure 6.8 right show some examples of our friction coefficient predictions. We observe that specular materials are more likely to have small coefficients of friction, which is consistent with physical intuition.

6.3.5 Rendering with a Physically-based Renderer

To render high quality images with realistic appearances, it is necessary to use a physically based renderer that models complex light transport effects such as global illumination and soft

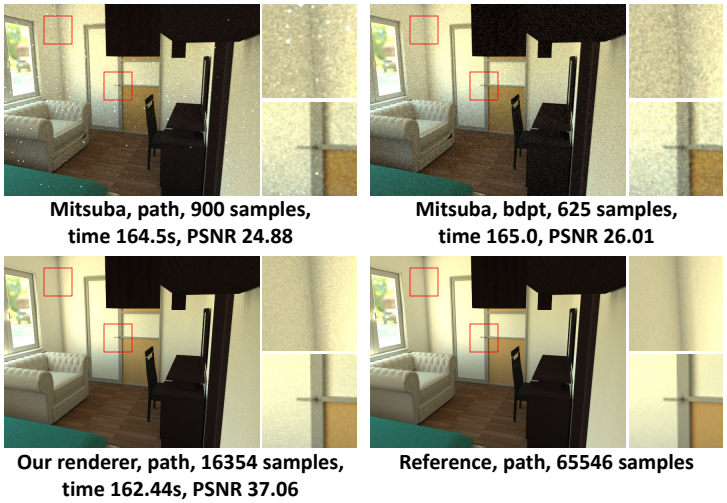


Figure 6.9. Comparisons of images rendered with Mitsuba and our GPU renderer in the same amount of time using path tracing. The quality of the image rendered by our renderer in less than three minutes is much better. It takes about 50 minutes for Mitsuba to achieve similar results.

shadows. However, current open source CPU renderers are too slow for creating a large dataset, especially to render per-pixel environment map. Thus, we implement our own physically-based GPU renderer using Nvidia OptiX [157], which models interreflection up to 7 bounces. To render a 480×640 image with 16384 samples per pixel, our renderer on Tesla V100 GPU needs 3-6 minutes, while Mitsuba on 16 cores of Intel i7-6900K CPU needs around 1 hour. Figure 6.9 compares images rendered with Mitsuba [89] and with our renderer using the same amount of time. Our renderer support ground-truth per-light contribution.

View selection

ScanNet provides the camera pose of each RGBD image. However, their distribution is biased towards views close to the scene geometry, to optimize scanning. On the contrary, we prefer views covering larger regions, matching typical human viewing conditions. To achieve this, we first sample different views along the wall, facing the center of the room. For each view, we render its depth and normal maps. Let $\mathbf{D}_i$ and $\mathbf{N}_i$ be the depth and normal of pixel i, $\mathbf{Grad}(\mathbf{N}_i)$ be the sum of absolute gradients of the normal in the three channels. We choose the view based

on computing a score defined as

$$\sum_i \mathbf{Grad}(\mathbf{N}_i) + 0.3\log(\mathbf{D}_i + 1). \tag{6.1}$$

Views with higher scores are used to create the dataset. An example of our view selection results is shown in Figure 6.2 (bottom right).

Other renderers

While our renderer will be publicly released, our assets (geometry, material maps, lights) are in a standard graphics format that could be used in other rendering environments. For example, common real-time rasterization engines like Unity or Unreal could be used for applications (such as robotics) which prefer real-time performance and do not require fully accurate global illumination. Furthermore, our per-pixel spatially-varying lighting maps could be used as high-quality precomputed lighting probes for photorealistic real-time rendering [135].

6.3.6 OpenRooms Dataset Statistics

Scene, image, semantic label distribution

We pick 1,287 of the 1,506 ScanNet scenes to instantiate our dataset, discarding those which cover very small portions of rooms. We randomly choose 1,178 scenes for training and 109 scenes for validation. For each scene, we choose views using our view selection method. For each rendered image, we render two others with different materials and lighting, as shown in Figure 6.2 (bottom-left). We render 118,233 HDR images at 480×640 resolution, with 108,159 in the training set and 10,074 in the validation set. We render semantic labels of all 44 classes of CAD models in OpenRooms. The distributions of scene categories and images, number of objects per class and the percentage of pixels per class are summarized in Figure 6.10. Note that the class distribution follows that of real scans in ScanNet indoor scenes.

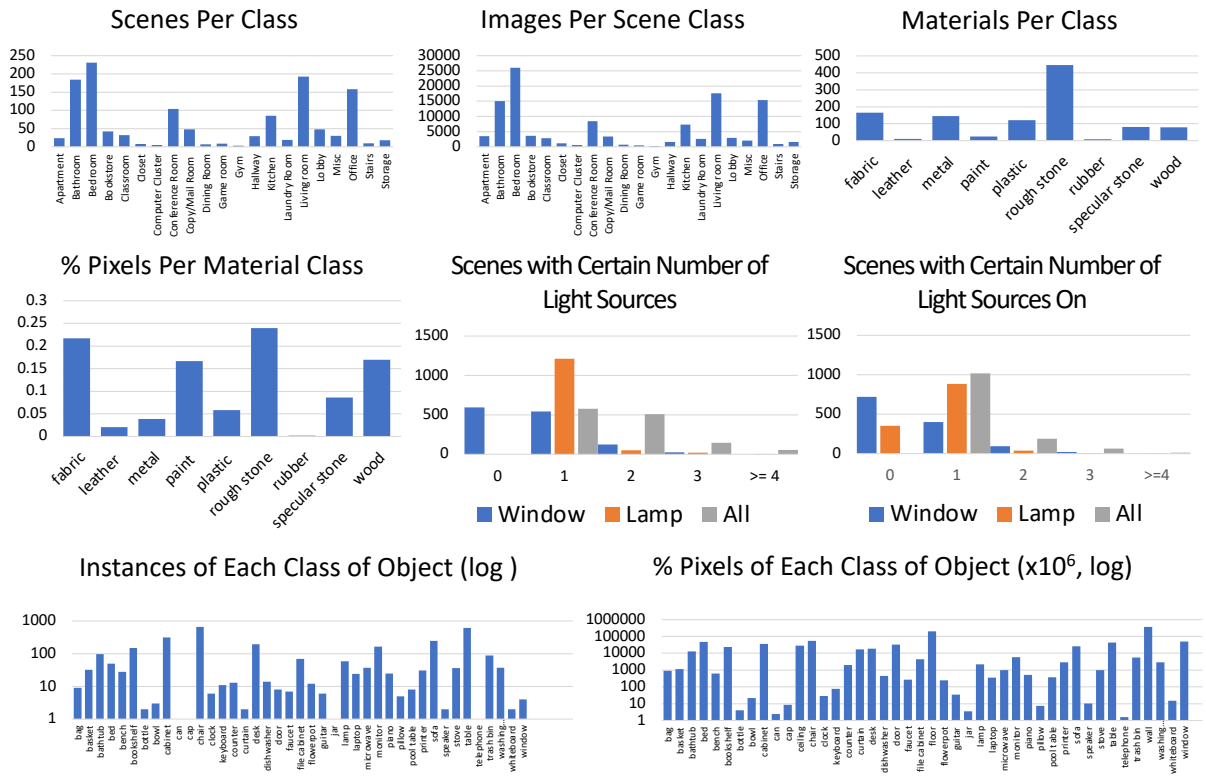


Figure 6.10. Dataset statistics for scene categories, images, materials, lighting and semantic labels.

Material distribution

We use 1,075 SVBRDFs from [3] to build OpenRooms, corresponding to the 9 categories shown in Figure 6.3. The number of materials per-category and their pixel distributions are summarized in Figure 6.10.

Lighting distribution

Figure 6.10 shows the distribution of the two types of light sources (windows and lamps). Each image has at least one light source "on" for rendering. For all the 118K images, we render spatially-varying environment maps and shading, with direct illumination only and with combined direct and indirect illumination. Moreover, we provide a parameterized representation for every visible and invisible light source, as well as render their individual direct shading contribution and visibility map. Compared to all prior works, OpenRooms provides significantly

Figure 6.11. Synthetic scene reconstruction results using scanned indoor scenes from SUNRGBD dataset. We visualize the reconstructed scenes rendered from different views with different material assignments.

more extensive and detailed supervision for complex lighting, which may allow new applications such as light source detection and editing.

Asset cost

Almost all the assets used for creating our dataset are publicly available and free for research use. The only non-free (but also publicly available) assets are the raw material maps from Adobe Stock [3] that cost less than US$500, while the material parameters annotated with our scenes are freely available. Note that photorealistic appearances may also be achieved using our tools with freely available materials, such as Substance Share [2] in Figure 6.2.

Dataset creation time

It takes 30s to annotate one scene layout and 1 minute to label materials for one object, leading to 64 hours for labeling the whole dataset, which was accomplished by students with knowledge of computer vision. Almost all rendering time is spent to render images and spatially-varying per-pixel environment maps, which takes 600s and 100s per image, respectively, for our custom renderer on a single 2080Ti GPU. In principle, we can render the dataset in 1 month using 40 GPUs.

6.3.7 Generalization to Another Dataset

To demonstrate that our framework can generalize to other datasets, we present our scene reconstruction results based on scanned indoor scenes from the SUNRGBD dataset. Unlike ScanNet [51], SUNRGBD only contains partial scans of the rooms with extremely incomplete and sparse point clouds. Moreover, unlike Scan2CAD [8], SUNRGBD only has 3D bounding box annotations for furniture locations and lacks full poses. Using this as initialization, we adjust the pose of the CAD models by simply using grid search to minimize the chamfer distance between the CAD model and the point cloud in the bounding box. Then we assign appropriate materials and lighting to the CAD models, as described before. In our experience, differing qualities of scans need to be addressed for geometry creation in different datasets, but our material and lighting mapping transfer across datasets with minimal effort. In Figure 6.11, we visualize the reconstruction results for SUNRGBD by rendering the created scenes from different viewpoints, with different material assignments. The rendered images present diverse appearances with plausible material and lighting assignments, with complex visual effects such as soft shadows and specularity being correctly handled.

6.4 Applications

In this section, we will demonstrate the scene understanding and robotic applications enabled by our OpenRooms framework and focus inverse rendering tasks in the next two chapters.

Light source detection

We use a ResNeXt101 [221] and FPN [120] pretrained model from Detectron2 [217] to train an instance segmentation network for light source detection (windows and lamps). We evaluate on OpenRooms and NYUv2 [188]. As shown in Table 6.2 and Figure 6.12, training on OpenRooms boosts accuracy on NYUv2 testing by around 5%, for both bounding box regression and segmentation.

Table 6.2. Bounding box regression and mask AP on OpenRooms and NYU2 [188] for light source (windows and lamps) detection.

Test on	OpenRoom		NYU2			
Train on OR/NYU2	Yes/ No		No/ Yes		Yes/ Yes	
	bbox	seg	bbox	seg	bbox	seg
AP(0.5:0.95)	80.2	70.1	17.1	15.3	23.5	21.6
AP-windows	85.8	63.2	11.9	12.7	20.5	20.6
AP-lamp	74.7	76.9	22.2	18.0	26.6	22.7

Figure 6.12. Light source detection on OpenRooms (OR) and NYUv2 [188]. Windows are better detected with OR training.

Semantic segmentation

We use DeepLabV3 [42] and PSPNet(50) [239] to pre-train semantic segmentation models on OpenRooms, then finetune and evaluate on NYUv2 [188] with 40 labels [72]. We also compare the results pre-trained on InteriorNet [111] with the same number of training images. As shown in Table 6.3 and Figure 6.13, results are comparable for the two models and register improvements with greater number of images for the two pre-training datasets.

Robotics and Embodied Vision

To facilitate research in robotics and embodied AI, OpenRooms supports transforming a rich 3D indoor scene model into an interactive environment, with realistic physical simulation through PyBullet [1]. A URDF file describe physical properties, such as mass and friction coefficients, for CAD models. This feature of OpenRooms establishes direct connections between appearance and physical properties of the environment, to provide a learning testbed for a range of topics including physics understanding from perception and policy generalization across environment and configuration changes. As an example, Figure 6.14 shows a classroom scenario where a robot from the iGibson dataset [185] is inserted into the scene and may perform

Figure 6.13. Semantic segmentation on OpenRooms and NYUv2 [188] using PSPNet(50) [239] and DeepLabV3 [42].

Table 6.3. Semantic segmentation trained on OpenRoom (OR) and InteriorNet (IN) [111] and fine-tuned on NYU [188] with PSPNet and DeepLabV3, using different number of images.

	PSPNet(50) [239]				DeepLabV3 [42]			
	mIoU		mAcc		mIoU		mAcc	
	10K	50K	10K	50K	10K	50K	10K	50K
IN	41.1	41.2	53.3	53.4	41.7	42.2	53.6	54.4
OR	40.8	41.1	53.0	52.5	42.5	42.9	54.5	55.1

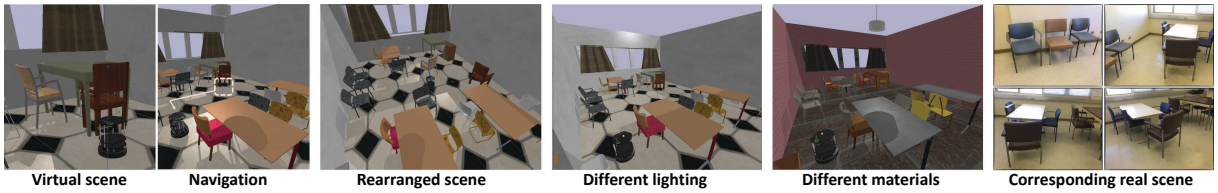


Figure 6.14. OpenRooms is integrated with a physics engine to create virtual scenes for robotics, potentially enabling studies for navigation and rearrangement across varying lighting and material, with possible correspondence to real scenes.

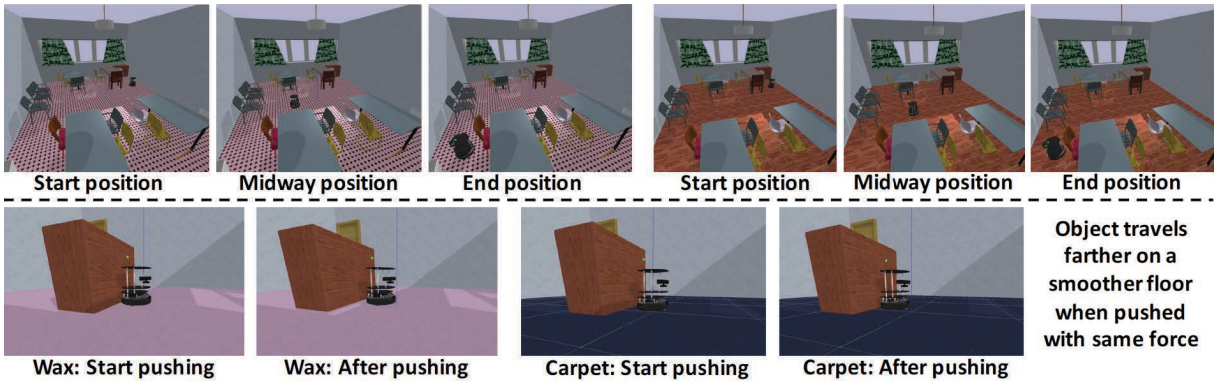


Figure 6.15. OpenRooms enables novel studies in navigation and rearrangement with material and lighting variations.

a navigation task. Furniture in the scene can be rearranged, while the lighting and material properties can also be changed. In Figure 6.15, we show navigation and rearrangement where different frictions of coefficient for the same scene lead to different pushing outcomes. Since we

create the scene from scans, correspondence is available to real scenes, which may be useful for sim-to-real transfer studies [92].

6.5 Conclusion

We have proposed methods that enable user-generated photorealistic datasets for complex indoor scenes, starting from existing public repositories of 3D scans, shapes and materials. We illustrate the process on over 1000 indoor scenes from ScanNet. In contrast to prior works, we provide high-quality ground truth for complex materials and spatially-varying lighting, including direct and indirect illumination, light sources, per-pixel environment maps and visibility. We also show our dataset can be integrated with physics engines and provide friction coefficients, which suggest interesting future studies in navigation, rearrangement and sim-to-real transfer. Our dataset and all tools used for its creation are publicly released.

In the next two chapters, we will introduce inverse rendering methods for indoor scenes built based on the OpenRooms dataset. We will start with reconstructing SVBRDF and SV-lighting of indoor scenes in Chapter 7 and then move one-step forward in Chapter 8, reconstructing the physically-based light sources to enable a full relighting of an indoor scene from a single image.

Chapter 6 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021 ("OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets", Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, Manmohan Chandraker). The dissertation author was the primary investigators and author of this paper.

Chapter 7

Indoor Scene Shape, SVBRDF and SV-Lighting Reconstruction from a Single Image

7.1 Introduction

In this chapter, we propose a deep inverse rendering framework for indoor scenes. From a single RGB image of an arbitrary indoor scene, we obtain a complete scene reconstruction, estimating shape, spatially-varying lighting, and spatially-varying, non-Lambertian surface reflectance. This is a significant advance compared to object-level appearance capturing method proposed in Chapter 4, because indoor scene present much more complex and diverse spatially-varying lighting and material appearance. Our high-quality reconstruction enables a range of photorealistic scene editing tasks. For example, in Figure 7.1 and Figure 7.2, we use our reconstruction to enable photorealistic virtual object insertion in a real image. Note how the inserted glossy spheres have realistic shading, shadowing due to scene occlusions and even reflections from the scene.

Training deep neural networks requires large-scale, labeled training data. While datasets of real-world geometry exist [51, 35], capturing real-world lighting and reflectance at scale is non-trivial. We use the OpenRooms dataset built in Chapter 6. As have been discussed in Chapter 6, OpenRooms is currently the only dataset that provides a large number of photorealistic images

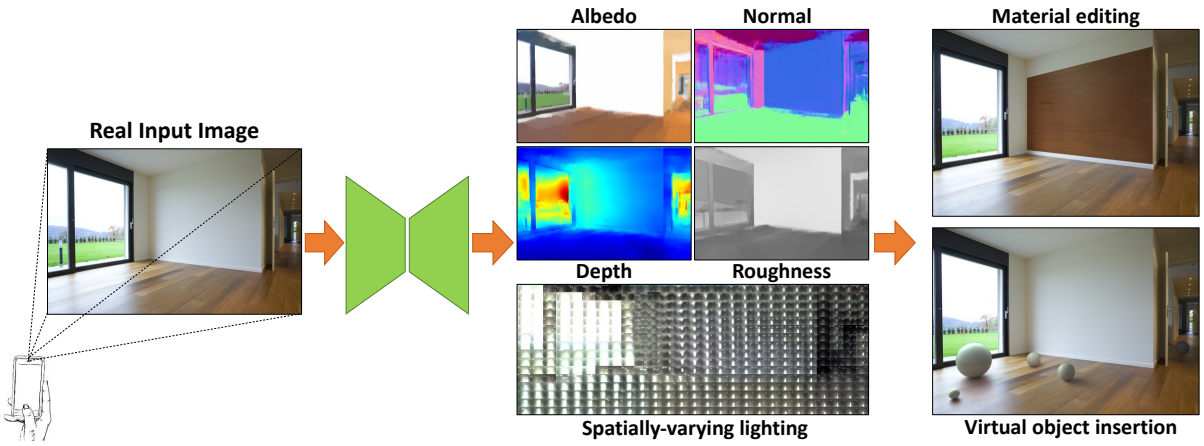


Figure 7.1. Given a single image of an indoor scene, we recover its diffuse albedo, normals, specular roughness, depth and spatially-varying lighting. By incorporating physical insights into deep learning, our high-quality predictions support applications like object insertion and material editing, even for specular materials and in real images. Note the completely shadowed sphere on the extreme right.

Figure 7.2. Photorealistic object insertion enabled by our indoor scene SVBRDF and SV-lighting prediction framework. The accompanying video [4] shows that our framework achieves close to spatially-consistent lighting prediction even without any smoothness prior.

with high-quality ground truth spatially-varying materials and lighting, making it the best choice for solving our indoor scene inverse rendering challenge.

An inverse rendering network would have to learn a model of image formation. The forward image formation model is well understood, and has been used in simple settings like

planar scenes and single objects [55, 118, 117, 121]. Indoor scenes are more complicated and exhibit challenging light transport effects like occlusions and inter-reflections. We address this by using a local lighting model—spatially-varying spherical gaussians (SVSGs). This bakes light transport effects directly into the lighting and makes rendering a purely local computation. We leverage this to design a fast, differentiable, *in-network* rendering layer that takes our geometry, SVBRDFs and SVSGs and computes radiance values. During training, we render our predictions and backpropagate the error through the rendering layer; this fixes the forward model, allowing the network to focus on the inverse task.

To the best of our knowledge, our work is the first demonstration of scene-level inverse rendering that truly accounts for complex geometry, materials and lighting, with effects like inter-reflections and shadows. Previous methods either solve a subset of the problem or rely on simplifying assumptions (Table 7.1). Despite tackling a much harder problem, we obtain strong results on the individual tasks. Most important, by truly decomposing a scene into physically-based scene factors, we enable novel capabilities like photorealistic 3D object insertion and scene editing in images acquired in-the-wild. Figure 7.2 shows object insertion examples on real indoor images, where our method achieves superior performance compared to [12, 61, 63]. Figure 7.1 shows a material editing example, where we replace the material of a surface in a real image, while preserving spatially-varying specular highlights. Such visual effects cannot be handled by previous intrinsic decomposition methods.

7.2 Related Work

The problem of reconstructing shape, reflectance, and illumination from images has a long history in vision. It has been studied under different forms, such as intrinsic images (reflectance and shading from an image) [15] and shape-from-shading (shape, and sometimes reflectance, from an image) [79]. Here, we focus on *single* image methods.

Table 7.1. A summary of scene-level inverse rendering. Karsch14's parametric lights cannot handle effects like shadowing [99]. Gardner17 [61] and Sengupta19 [181] predict a single lighting for the scene, thus, cannot handle spatial variations. Li18's shading entangles geometry and lighting [114]. Barron13 uses RGBD input and non-physical image formation [12]. Azinović19 [10] needs multiple images with 3D reconstruction as input. Our spherical Gaussians representation for local lighting is demonstrably better than spherical harmonics in Barron13 [12], Sengupta19 [181] and Garon19 [63]. Song19 [189] and several others do not handle complex SVBRDF.

	Karsch 2014	Barron 2013	Eigen 2015	Gardner 2017	Li 2018	LeGendre 2019	Azionvić 2019	Garon 2019	Song 2019	Sengupta 2019	Ours
Geometry	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓
Reflectance	Diffuse	Diffuse	✗	✗	Diffuse	✗	Microfacet	✗	✗	Phong	Microfacet
Lighting	Local	Local	✗	Global	✗	Global	Local	Local	Local	Global	Local

Single objects

Many inverse rendering methods focus on reconstructing single objects. Even this problem is ill-posed and many methods assume some knowledge of the object in terms of known lighting [154, 91] or geometry [123, 175]. Recent methods have leveraged deep networks to reconstruct complex SVBRDFs from single images of planar scenes [55, 117], objects of a specific class [121] or homogeneous BRDFs [136]. Other methods address illumination estimation [64]. We tackle the much harder case of large-scale scene modeling and do not assume scene information. Barron and Malik [13] propose an optimization-based approach with hand-crafted priors to reconstruct shape, Lambertian reflectance, and distant illumination from a single image. In Chapter 4, we tackle the same problem with a deep network and an object-specific rendering layer. Extending these methods to scenes is non-trivial because the light transport is significantly more complex.

Indoor scenes

Previous work recognizes materials in indoor scenes [19] and decomposes indoor images into reflectance and shading layers [17, 114]. Techniques have also been proposed for single image geometric reconstruction [57] and lighting estimation [78, 61]. Those methods estimate only one scene factor. Barron and Malik [12] reconstruct Lambertian reflectance and spatially-

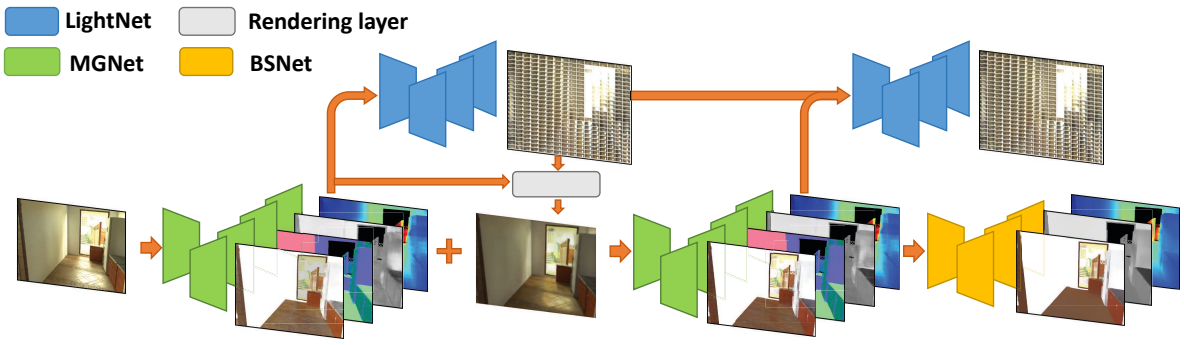


Figure 7.3. Our network design consists of a cascade, with one encoder-decoder for material and geometry prediction and another one for spatially-varying lighting, along with a physically-based differentiable rendering layer and a bilateral solver for refinement.

varying lighting but require RGBD input. Karsch et al. [98] estimate geometry, Lambertian reflectance and 3D lighting, but rely on extensive user input to annotate geometry and initialize lighting. An automatic, rendering-based optimization is proposed in [99] to estimate all these scene factors, but using strong heuristics that are often violated in practice. Recent deep networks also do not account for either spatially-varying lighting [181] or complex SVBRDF [242]. In contrast to all those methods, our network learns to predict geometry, complex SVBRDFs and spatially-varying lighting in an end-to-end fashion.

7.3 Network Design

Estimating spatially-varying lighting, complex SVBRDF and geometry from a single indoor image is an extremely ill-posed problem, which we solve using priors learned by our physically-motivated deep network (architecture shown in Figure 7.3). Our network consists of cascaded stages of a SVBRDF and geometry predictor, a spatially-varying lighting predictor and a differentiable rendering layer, followed by a bilateral solver for refinement.

Material and geometry prediction

The input to our network is a single gamma-corrected low dynamic range image **I**. Inspired by Chapter 3 and 4, we use a single encoder to capture correlations between material

and shape parameters, obtained using four decoders for diffuse albedo (**A**), roughness (**R**), normal (**N**) and depth (**D**). Skip links are used for preserving details. Then the initial estimates of material and geometry are given by

$$\mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D} = \mathbf{MGNet}_0(\mathbf{I}). \tag{7.1}$$

Spatially Varying Lighting Prediction

Inverse rendering for indoor scenes requires predicting spatially varying lighting for every pixel in the image. Directly using an environment map as the lighting representation leads to a very high dimensional output space, that causes memory issues and unstable training due to small batch sizes. Spherical harmonics are a compact lighting representation that have been used in recent works [94], but do not efficiently recover high frequency lighting necessary to handle specular effects [167, 16]. Instead, we follow pre-computed radiance transfer methods [204, 70, 223] and use isotropic spherical Gaussians that approximate all-frequency lighting with a smaller number of parameters. As have been introduced in Chapter 2, we can model the lighting as a spherical function $\mathbf{L}(\omega)$ approximated by the sum of spherical Gaussian lobes:

$$\mathbf{L}(\omega) = \sum_{n=1}^{N} \mathbf{w}_n \exp(\lambda_n(\mathbf{d}_n \omega - 1)) \tag{7.2}$$

where ω and $\{\mathbf{d}_n\}$ are vectors on the unit sphere, $\{\mathbf{w}_n\}$ controls RGB color intensity and $\{\lambda_n\}$ controls the bandwidth.

Each spherical Gaussian lobe is represented by 6 dimensional parameters $\{\mathbf{w}_n, \lambda_n, \mathbf{d}_n\}$. Figure 2.4 compares the images rendered with a 12-spherical Gaussian lobes approximation (72 parameters) and a fourth-order spherical harmonics approximation (75 parameters). It is evident that even using fewer parameters, the spherical Gaussian lighting performs better, especially close to specular regions.

Our novel lighting prediction network, $\mathbf{LightNet}_0(\cdot)$, accepts predicted material and geometry as input, along with the image. It uses a shared encoder and separate decoders to

predict $\{\mathbf{w}_n\}, \{\lambda_n\}, \{\mathbf{d}_n\}$.

$$\{\mathbf{w}_n\}, \{\lambda_n\}, \{\mathbf{d}_n\} = \mathbf{LightNet}_0(\mathbf{I}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D}). \tag{7.3}$$

Our predicted lighting is HDR, which is important for applications like relighting and material editing.

Differentiable rendering layer

Our OpenRooms dataset in Chapter 6 provides ground truth for all scene components. But to model realistic indoor scene appearance, we additionally use a differentiable in-network rendering layer to mimic the image formation process, thereby weighting those components in a physically meaningful way. We implement this layer by numerically integrating the product of SVBRDF ρ and spatially-varying lighting $\mathbf{L}$ over the hemisphere. Let $\omega_{jj'} = \omega(\phi_j, \theta_{j'})$ be a set of light directions sampled over the upper hemisphere, with ω_o the view direction. The rendering layer computes diffuse $\mathbf{I}_d$ and specular images $\mathbf{I}_s$ as:

$$\mathbf{I}_d = \sum_{j,j'} \rho_a(\omega_o, \omega_{jj'}; \mathbf{A}) \mathbf{L}\left(\omega_{jj'}\right) \cos\theta_j \mathrm{d}\omega, \tag{7.4}$$

$$\mathbf{I}_s = \sum_{j,j'} \rho_s(\omega_o, \omega_{jj'}; \mathbf{R}) \mathbf{L}\left(\omega_{jj'}\right) \cos\theta_j \mathrm{d}\omega, \tag{7.5}$$

where $\mathrm{d}\omega$ is the differential solid angle. ρ_a and ρ_s are the Lambertian term and the specular term of the microfacet BRDF model defined in (2.6). We sample 16×8 lighting directions. While this is relatively low resolution, we empirically find, as shown in Figure 2.4, that it is sufficient to recover most high frequency lighting effects.

Loss Functions

Our loss functions incorporate physical insights. We first observe that two ambiguities are difficult to resolve: the ambiguity between color and light intensity, as well as the scale ambiguity of single image depth estimation. Thus, we allow the related loss functions to be scale

invariant. For material and geometry, we use the scale invariant L_2 loss for diffuse albedo ($\mathscr{L}_a$), L_2 loss for normal ($\mathscr{L}_n$) and roughness ($\mathscr{L}_r$) and a scale invariant log-encoded loss for depth ($\mathscr{L}(d)$) due to its high dynamic range:

$$\mathscr{L}_d = \|\log(\mathbf{D}+1) - \log(c_d\hat{\mathbf{D}}+1)\|_2^2, \tag{7.6}$$

where c_d is a scale factor computed by least squares regression. For lighting estimation, we supervise with the final reconstructed environment maps ($\mathscr{L}_l$). We also add a a scale invariant L_2 rendering loss:

$$\mathscr{L}_{ren} = ||(\mathbf{I} - c_a\mathbf{I}_a - c_s\mathbf{I}_s)||_2^2 \tag{7.7}$$

where $\mathbf{I}_d$ and $\mathbf{I}_s$ are rendered using (7.4) and (7.5), respectively, while c_a and c_s are positive scale factors computed using least square regression. The final loss function is a weighted summation of the proposed losses:

$$k_a\mathscr{L}_a + k_n\mathscr{L}_n + k_r\mathscr{L}_r + k_d\mathscr{L}_d + k_l\mathscr{L}_l + k_{ren}\mathscr{L}_{ren} \tag{7.8}$$

Refinement using bilateral solver

We use an end-to-end trainable bilateral solver to impose a smoothness prior [14, 114]. The inputs include the prediction, the estimated diffuse albedo **A** as a guidance image and confidence map **C**. We train a shallow network with three sixteen-channel layers for confidence map predictions. Let $\mathbf{BS}(\cdot)$ be the bilateral solver and $\mathbf{BSNet}_X(\cdot)$ be the network for confidence map predictions where $\mathbf{X} \in \{\mathbf{A}, \mathbf{R}, \mathbf{D}\}$. We do not find refinement to have much effect on normals. The refinement process is:

$$\mathbf{C_X} = \mathbf{BSNet}(\mathbf{X}, \mathbf{I}), \quad X \in \{\mathbf{A}, \mathbf{R}, \mathbf{D}\} \tag{7.9}$$

$$\mathbf{X}^* = \mathbf{BS}(\mathbf{X}; \mathbf{C_X}, \mathbf{A}) \tag{7.10}$$

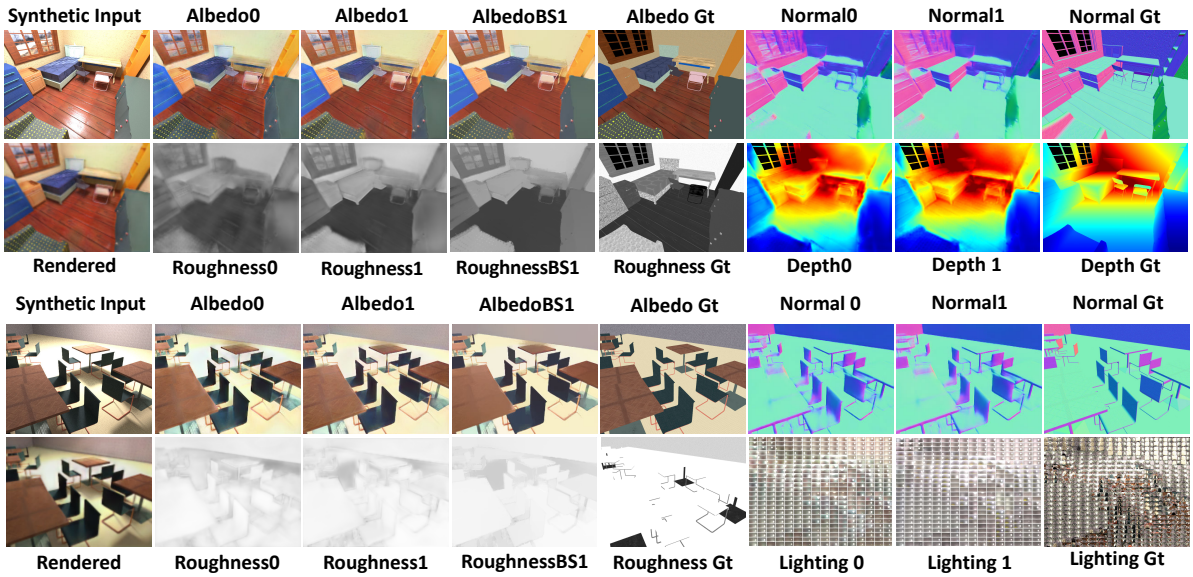


Figure 7.4. Results on synthetic images. Given a single image, our estimated albedo, normals, depth, roughness and lighting are close to ground truth. We observe that cascade structure helps remove noise and increase details, while bilateral solver can enhance smoothness prior.

where we use $(^*)$ for predictions after refinement.

Cascade Network

Akin to recent works on high resolution image synthesis [97, 43] and inverse rendering in Chapter 4, we introduce a cascaded network that progressively increases resolution and iteratively refines the predictions through global reasoning. We achieve this by sending both the predictions and the rendering layer applied on the predictions to the next cascade stages, $\mathbf{MGNet}_1(\cdot)$ for material and geometry and $\mathbf{LightNet}_1(\cdot)$ for lighting, so that the network can reason about their differences. Cascade stages have similar architectures as their initial network counterparts.

7.4 Experiments

We now conduct studies on the roles of various components in our pipeline, compare to prior works and illustrate applications such as high quality object insertion and material editing in real images that can only be enabled by our holistic solution to inverse rendering.

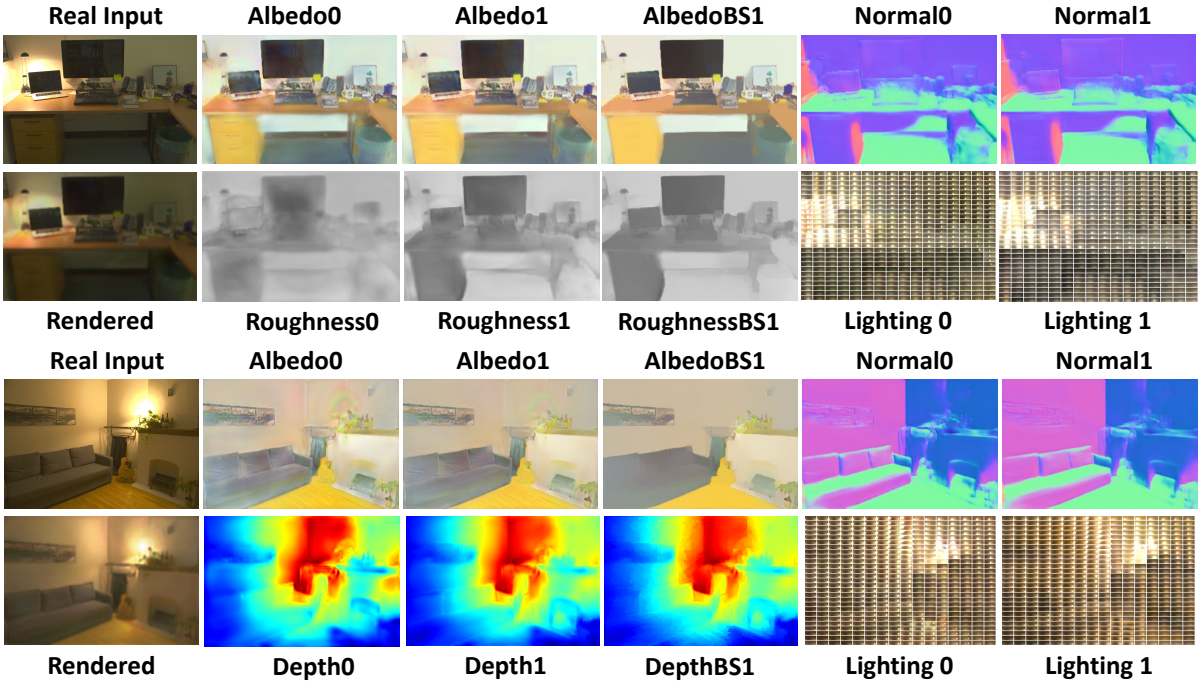


Figure 7.5. Results on real images, for single-image depth, normals, spatially-varying material and lighting. Improvements are observed due to the cascade structure and bilateral solver.

Table 7.2. Ablation study for the network architecture on our proposed dataset. We report the scale invariant L_2 loss for albedo (**A**), L_2 loss for normal (**N**), scale invariant log L_2 loss for depth (**D**), L_2 loss for roughness (**R**) and scale invariant $\log(x+1)$ L2 loss for per-pixel lighting (**L**). We observe both cascade structure and bilateral solver can improve the prediction accuracy.

	$A(10^{-3})$	$N(10^{-2})$	$D(10^{-2})$	$R(10^{-2})$	L
Cascade0	9.99	4.51	5.18	6.59	0.150
Cascade1	9.43	4.42	4.89	6.64	**0.146**
Bilateral solver	**9.29**	-	**4.86**	**6.57**	-

7.4.1 Analysis of Network and Training Choices

We study the effect of the cascade structure, joint training and refinement. Quantitative results for material and geometry predictions on the proposed dataset are summarized in Table 7.2, Figure 7.4 and Figure 7.5

Cascade

The cascade structure leads to clear gains for shape, BRDF and lighting estimation by iteratively improving and upsampling our predictions in Tables 7.2 and Figure 7.4. This holds

Table 7.3. Left: intrinsic decomposition on IIW dataset [18]. Right: normal and depth prediction on NYU dataset [188].

	Training	WHDR↓
Ours	Ours + IIW	16.4
Li18[114]	CGI + IIW	17.5
Sen.19[181]	CGP + IIW	16.7

Method	Mean(°)↓	Med.(°)↓	Depth↓
Ours	25.3	18.0	0.171
Sen.19[181]	21.1	16.9	–
Zhang17[238]	21.7	14.8	–

for real data too, as shown in Figure 7.5. We observe that the cascade structure can effectively remove noise and preserve high frequency details for both materials and lighting. The errors in our shape, material and lighting estimates are low enough to photorealistically edit the scene to insert new objects, while preserving global illumination effects as shown in Figure 7.2.

Refinement

Finally, we study the impact of the bilateral solver. Quantitative improvements over the second cascade stage in Table 7.2 are modest, which indicates that the network already learns good smoothness priors by that stage. But we find the qualitative impact of the bilateral solver to be noticeable on real images (for example, diffuse albedo in Figure 7.5), thus, we use it in all our real experiments.

Qualitative examples

In Figure 7.4 and Figure 7.5, we use a single input image from our synthetic test set to demonstrate depth, normal, SVBRDF and spatially-varying lighting estimation. The effectiveness is further illustrated by low errors with respect to ground truth Figure 7.4.

7.4.2 Comparisons with Previous Works

We address the problem of holistic inverse rendering with spatially-varying material and lighting which has not been tackled earlier. Yet, it is instructive to compare our approach to prior ones that focus on specific sub-problems.

Figure 7.6. Comparisons of object insertion on real images of Garon et al. [63]. Our overall appearances look more realistic. For example, note the bunny under bright light (top right) in the top row and in the shadow (bottom middle) in bottom row. Also see Table 7.4.

Figure 7.7. Material editing on real images. Left is the original image and right is the rendered one with the material replaced in a part of the scene. We observe that the edited material looks photorealistic and even high frequency details from specular highlights and spatially-varying lighting are rendered well.

Intrinsic decomposition

We compare our method with two prior works Li18 [114] and Sen.19[181] on the IIW dataset [17] for intrinsic decomposition evaluation. The results are tabulated in Table 7.3, left. Our method is comparable to prior state-of-the-art based on artist-created SUNCG dataset [191].

Depth and normal estimation

We fine-tune our network, trained on our synthetic dataset, on NYU dataset [188]. The test error on NYU dataset is summarized in Table 7.3, right. Zhang et al. [238] achieve state-of-the-art performance for normal estimation using a more complex fine-tuning strategy and

Table 7.4. User study on object insertion indicating the % of pair-wise comparisons where human annotators thought we outperformed an alternative method; we outperform all prior methods.

	Barron13 [12]	Gardner17 [61]	Garon19 [63]	Li20 [115]
Ours vs.	88.19%	66.16%	56.53%	54.77%

with more than six times as much training data. Although we do not achieve state-of-the-art performance on this task, it's not our main focus. Rather, we aim to show the wide utility of our proposed dataset and demonstrate estimation of factors of image formation good enough to support photo-realistic augmented reality applications.

Object insertion

Given a single real image, we insert a novel object with photorealistic shading, specularity and global light transport effects. This is a crucial ability for high quality augmented reality applications. To simplify the demonstration, we estimate the shape, material and lighting using our cascade network, then select a planar region of the scene to insert an object. We relight the object using the estimated lighting. It may be observed on qualitative examples in Figures 7.1), 7.2, 7.6 (all containing real images) that even complex visual effects such as shadows and reflections from other parts of the scene are faithfully rendered on the inserted object.Further, [63] provides a dataset of 20 real indoor images with ground truth spatially-varying lighting. For each image, we render a virtual bunny into the scene lit by ground-truth or predicted lighting (Figure 7.6). We also performed an AMT user study on these images. Following the protocol in [63], users are shown image pairs, and asked to pick which is more realistic. As shown in Tab. 7.4, we outperform prior methods by a large margin.

Material Editing

Editing material properties of a scene using a single photograph has applications for interior design and visualization. Our disentangled shape, material and lighting estimation allows rendering new appearances by replacing materials and rendering using the estimated lighting. In Figure 7.7 (all real images), we replace the material of a planar region with another kind of

material and render the image using the predicted geometry and lighting, whose spatial variations are clearly observable. We can see the specular highlight in the is rendered after changing to specular materials. This is not possible for intrinsic decomposition methods, which cannot determine incoming lighting direction.

7.5 Conclusion

We have presented the first holistic inverse rendering framework that estimates disentangled shape, SVBRDF and spatially-varying lighting, from a single image of an indoor scene. Insights from computer vision, graphics and deep convolutional networks are utilized to solve this challenging ill-posed problem. A GPU-accelerated renderer is used to synthesize a large-scale, realistic dataset with complex materials and global illumination. Our per-pixel SVSG lighting representation captures high frequency effects. Our network imbibes intuitions such as a differentiable rendering layer, which are crucial for generalization to real images. Design choices such as a cascade structure and a bilateral solver lead to further benefits. Despite solving the joint problem, we obtain strong results on various sub-problems, which highlights the impact of our dataset, representations and network. We demonstrate object insertion and material editing on real images that capture global illumination effects, motivating applications in augmented reality and interior design.

However, the framework proposed in this chapter cannot be used for indoor scene light editing, such as turning off a lamp or open a virtual window. The essential reason is because our (SVSG) spatially-varying lighting representation does not separate direct lighting and global illumination. While this facilitate virtual object insertion, it is not sufficient for scene editing applications with non-local changes, such as light editing, where the global illumination needs to be modeled and edited explicitly. We will present our solution in the next chapter.

Chapter 7 is based on the material as it appears in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020 (“Inverse Rendering for Complex Indoor Scenes: Shape,

Spatially-Varying Lighting and SVBRDF from a Single Image", Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker). The dissertation author was the primary investigators and author of this paper.

Chapter 8

Indoor Scene Light Sources Reconstruction and Editing with Complex Light Transport

8.1 Introduction

Light sources of various shapes, colors and types, such as lamps and windows, play an important role in determining indoor scene appearances. Their influence leads to several interesting phenomena such as light shafts through a window on a sunlit day, highlights on specular surfaces due to incandescent lamps, interreflections from colored walls, or shadows cast by furniture in the room. Correctly attributing those effects to individual visible or invisible light sources in a single image enables abilities for photorealistic augmented reality that have previously been intractable — virtual furniture insertion under varying illuminations with consistent highlights and shadows, virtual try-on of wall paints with accurate global interreflections, or morphing a room under fluorescent lights into one reflecting the sunrise through a window (Figure 8.1).

In Chapter 7, we estimate *lighting* in indoor scenes, but achieving the above outcomes requires estimating and editing *light sources*. While both are highly ill-posed for single-image inputs, we posit that the latter presents fundamentally different and harder challenges. First, it requires disentangling the individual contributions of both visible and invisible light sources, independent of the effects of geometry and material. Second, it requires reasoning about

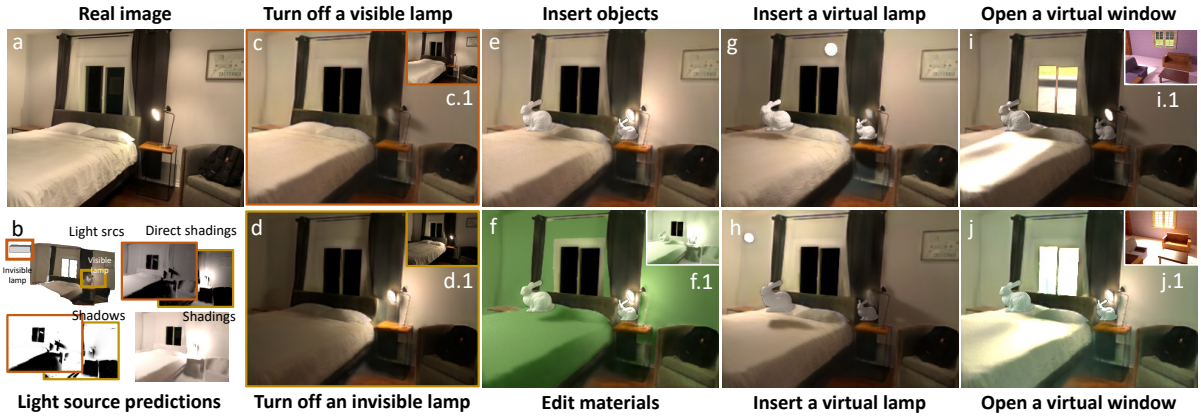


Figure 8.1. We present the first method that can edit indoor lighting from a single LDR image. Given the input (a), our framework first estimates physically-based light source parameters, for both visible and invisible lights, and then renders their direct contributions and interreflections through a neural rendering framework (b). Our framework can turn off visible and invisible light sources (c and d) with results that closely match the ground truths (c.1 and d.1). It can insert virtual objects at arbitrary locations (e) with consistent changes of highlight and shadow and edit materials with color bleeding being correctly rendered, as shown in the rendered image (f) and shading (f.1). It can also insert virtual lamps (g and h) and open a virtual window to let sunlight (i.1 and j.1) shine into the room.

long-range effects such as interreflections, shadows and highlights, while also being precise about highly localized 3D shapes, spectra, directions and bandwidths of light sources, where minor errors can lead to global artifacts due to the above distant interactions. Third, it requires photorealistic re-rendering of the scene despite only partial observations of geometry and material, while handling complex light transport.

We solve the above challenges by bringing together a rich set of insights across physically-based vision and neural rendering. Given a single LDR image of an indoor scene, with predicted depth map and masks for visible lights, we propose an inverse rendering method to estimate a parametric model of both visible and invisible light sources (in addition to a per-pixel SVBRDF). Beyond a 3D localization of light sources, our modeling accurately supports their physical properties such as geometry, color, directionality and fall-off. Next, we design a *neural differentiable renderer* that judiciously uses classical methods and learned priors to synthesize high-quality images from predicted reflectance and light sources. We accurately model long-range light

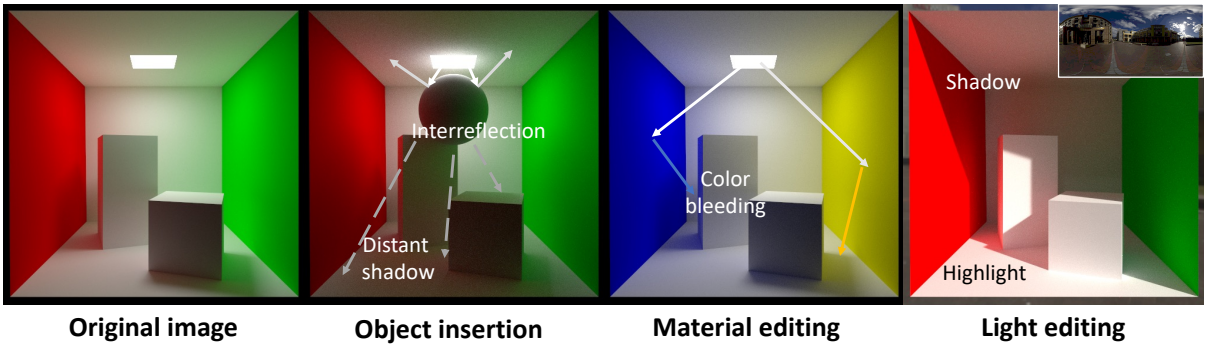


Figure 8.2. An illustration of desiderata for photorealistic single-image scene editing, where in contrast to prior works, global effects are handled by our method through explicit estimation of light sources. These include the global influence of inserted objects to create distant shadows and interreflections, edited materials leading to color bleeding on far surfaces, light shafts by opening a window or darkening of the room by turning off a lamp.

transport through a physically-based Monte Carlo ray tracer with a learned shadow denoiser to render direct irradiance and visibility, which combines with an indirect irradiance network to predict local incoming lighting at every pixel. Our neural renderer injects the inductive bias of physical image formation in training, while allowing rendering and editing of global light transport from partial observations, as well as optimization to refine editing outputs.

Our parametric light source estimation and physically-based neural renderer allow *full scene relighting* with intuitive editing of multiple lamps and windows. In Figure 8.1(c,d), we turn off individual visible and invisible lamps. Beyond standard object insertion of prior works (e), we visualize inserted objects by "turning on" a new lamp (g,h) or "opening" a window with incoming sunlight (i,j). In each case, global effects such as highlights, shadows and interreflections are accurately created for the entire scene by the neural renderer, and are also properly handled when we edit material properties of scene surfaces (f). In the accompanying video [4], we show that these editing effects are consistent as we move virtual objects and light sources, or gradually change materials.

Table 8.1. Compared to prior works on inverse rendering, ours enables full scene relighting with global effects for inserted objects, edited materials or light sources. Also see Figs. 8.1 and 8.2.

	Input	Object insertion		Material editing		Light editing	
		Position	Non-local	Specular	Non-local	Lamp	Window
Auto, Karsch 14	Single	Any	✓	✗	✗	✓	✗
CGI, Li 18	Single	✗	✗	✗	✗	✗	✗
DeepPara, Gardner 19	Single	Any	✓	✗	✗	✗	✗
InvIndoor, Li 20	Single	Surface	✗	✓	✗	✗	✗
Lighthouse, Srinivasan 20	Stereo	Any	✗	✗	✗	✗	✗
FreeView, Philip 21	Multi.	✗	✗	✗	✗	✓	✗
Ours	Single	Any	✓	✓	✓	✓	✓

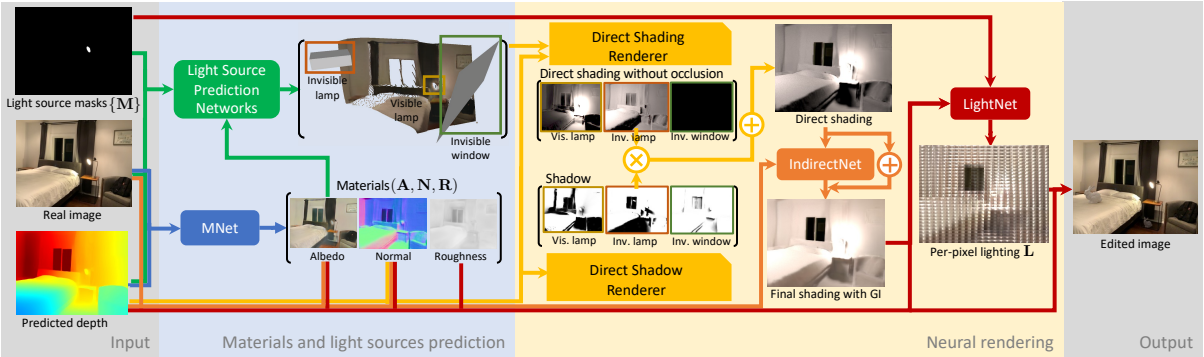


Figure 8.3. Overview of our method. We start from an RGB image. The depth map and visible light source masks can be estimated from the RGB image or given as additional inputs. We estimate per-pixel material parameters (albedo, normal, roughness) using a network (blue). Next, we estimate light sources (windows and lamps, visible and invisible) using four networks (green). At this point, we can edit the scene representation (lights, materials, depth). To render the edited representation back into an image, we use a neural renderer consisting of three modules: direct shading, shadow (yellow), and indirect shading module (orange). The result is per-pixel shading (diffuse irradiance), which can be turned into per-pixel lighting (a grid of incoming radiance environment maps) using another network (red).

8.2 Related Work

Lighting estimation and representation.

Many single image approaches estimate lighting globally as an environment map [52, 61, 109], which cannot express the complex spatial variation of indoor illumination. Some

recent works model spatial variations as per-pixel environment maps [13, 243, 63], including our method in Chapter 7, or volumes [193, 208]. However, these are non-parametric representations, which can mainly be used for object insertion, while we estimate editable windows and lamps (visible and invisible) with physically meaningful properties (such as position, direction, shape, size and intensity). Gardner et al. [62] predict a fixed number of spherical Gaussian lobes to approximate indoor light sources but do not handle light editing or its global effects. Zhang et al. recover the geometry and radiance of an empty room but cannot handle furniture inside [235]. Karsch et al. reconstruct geometry, reflectance and lighting but do not model windows and invisible scene contributions, require extensive user inputs [98] or face artifacts from imperfect heuristics or optimization [99]. In contrast, our physically-based neural renderer synthesizes photorealistic images with complex light transport, to enable relighting, light source insertion and removal from a single image.

Neural rendering and relighting.

NeRF [137] and other volumetric neural rendering approaches have achieved photo-realistic outputs, but usually limited to view synthesis [137, 232, 122]. A few recent works [23, 22, 27, 192, 220] handle relighting, but use a per-object optimization from a large set of images. Philip et al. [160] demonstrate relighting for outdoor scenes but require multiple images. Concurrent to our work, Philip et al. [161] consider indoor relighting, but require a large number of high-resolution RAW images, cannot reconstruct complex directional sunlight and do not support material editing and object insertion with their neural renderer. As shown in Figure 8.2 and Table 8.1, our modeling and neural rendering enable applications not possible for prior works, such as light source insertion and removal, or insertion of virtual objects and changing of materials with non-local effects, with a single LDR image.

8.3 Material and Light Source Prediction

Our overall framework is summarized in Figure 8.3. In this section, we describe our novel, physically meaningful and editable representations, while Sec. 8.4 describes our neural renderer that is differentiable with respect to light sources to facilitate training and editing of complex light transport.

Per-pixel normal and material prediction

We first train a U-net similar to [115] to predict material parameters per pixel of the input image: diffuse albedo (**A**), normal (**N**) and roughness (**R**), following the SVBRDF model of [95]. The input to the network is a 240×320 LDR image (**I**) and its corresponding depth map (**D**), which in our case can be predicted by a state-of-the-art monocular depth prediction network [169]. We predict the normals directly, instead of computing them as the normalized gradient of depth to avoid artifacts and discontinuities. Thus, our prediction is given by $\{\mathbf{A}, \mathbf{N}, \mathbf{R}\} = \mathbf{MNet}(\mathbf{I}, \mathbf{D})$.

8.3.1 Light Source Representation

To enable indoor scene relighting from a single LDR image, we need light source representations that are editable, expressive enough for different types of lighting and realistic enough for convincing rendering of complex scenes. We model the radiance and geometry of two types of common indoor light sources with very different properties: (a) *windows* that usually cover large areas and may induce strong directional lighting from the sun, and (b) *lamps* that tend to be small but with more complex geometry.

Radiance

The emitted radiance of lamps can be modeled by a standard Lambertian model, where every surface point with intensity **w** emits light uniformly into its hemisphere. However, the radiance distribution of windows can be strongly directional due to sunlight coming through on a clear day, which is important for capturing realistic indoor lighting but often neglected by prior methods [196, 161, 193]. A recent work [208] models directional lighting with a single

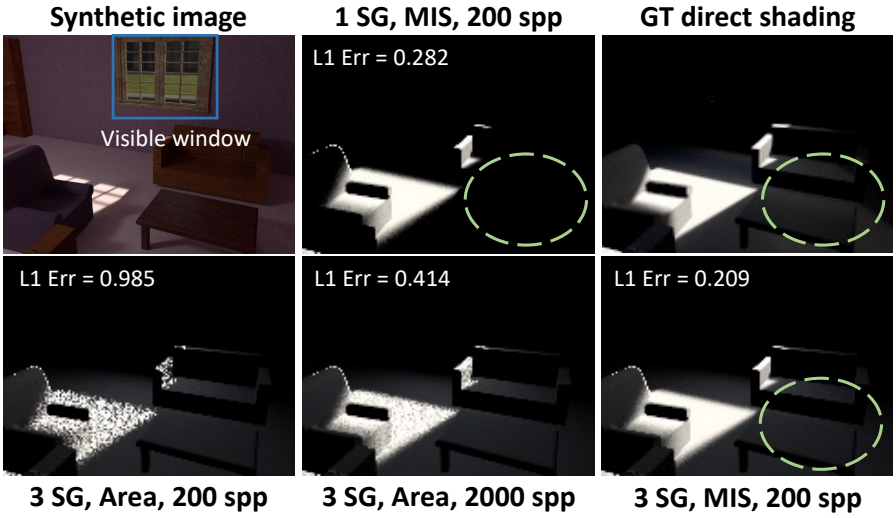


Figure 8.4. Comparisons of direct shading rendered from different window representations with different sampling methods. We show that our 3 SGs models ambient lighting much better than a single SG, as shown in the green circle, and MIS sampling leads to much less noise compared to sampling window area uniformly.

spherical Gaussian (SG), but as shown in Figure 8.4, cannot recover ambient effects leading to suboptimal rendering. Instead, we model the directional distribution of window radiance with 3 SGs corresponding to the sun, sky and ground. For a ray in direction ω that hits the window, its intensity is:

$$\mathbf{L}_{\mathtt{W}}(\omega) = \sum_{\mathbf{n}\in\{\text{sun, sky, grd}\}} \mathbf{w}_n \exp\left(\lambda_n(\mathbf{d}_n \cdot \omega - 1)\right). \tag{8.1}$$

Each SG is defined by three parameters $\mathbf{SG_n} = (\mathbf{w}_n, \lambda_n, \mathbf{d}_n)$, for intensity, bandwidth and direction of lighting. Figure 8.4 shows that our representation with multiple importance sampling leads to direct shading close to the ground-truth.

Geometry

Window geometry can be simply approximated by a rectangle $\{\mathbf{c}, \mathbf{x}, \mathbf{y}\}$, where $\mathbf{c}$ is the center and $\mathbf{x}, \mathbf{y}$ are the two axes. However, indoor lamps present more diverse geometry. Naively representing a lamp with a 3D bounding box $\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ works for invisible lamps in the scene, but it often leads to artifacts for visible lamps, as the imperfect shape generates incorrect highlights. Therefore, we carefully design a new visible lamp representation as shown in Figure 8.5. We first

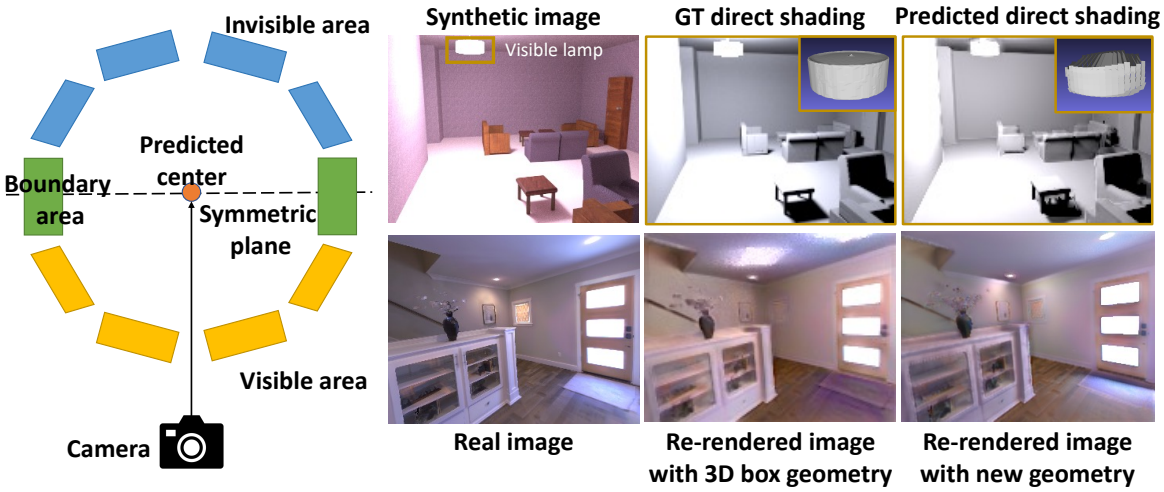


Figure 8.5. A demonstration of our visible lamp geometry representation. Our representation for visible lamps is much less likely to cause highlight artifacts and wrong shadows compared to a standard 3D bounding box.

identify the visible surface based on the depth $\mathbf{D}$ and lamp segmentation mask $\mathbf{M}_{\text{L}}$, reconstruct the invisible surface by reflecting the visible surface with respect to the lamp center $\mathbf{c}$ and then add the boundary area. As shown in Figure 8.5, our new representation can effectively constrain the lamp geometry and achieve realistic rendering without highlight artifacts for difficult real world examples.

8.3.2 Light Source Prediction

We use four neural networks to predict visible and invisible light sources for the lamp and window categories. For visible light sources, the inputs to the network include extra instance segmentation masks for visible lamps and windows that are turned on in the scene. We can obtain the instance segmentation mask by either fine-tuning a Mask R-CNN[75] for our dataset, combined with a graph-cut based post processing to refine the boundaries, or manually draw the masks. Let $\mathbf{M}_{\text{W}}$ be a mask for a window and $\mathbf{M}_{\text{L}}$ be a mask for a lamp. For each visible window

and lamp, we have

$$\{\mathbf{c}, \mathbf{w}\} = \textbf{VisLampNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}_{\mathtt{L}}),$$
$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{SG}_{\text{sun}}, \mathbf{SG}_{\text{sky}}, \mathbf{SG}_{\text{grd}}\} = \textbf{VisWinNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}_{\mathtt{W}}).$$

We assume one invisible lamp as a 3D bounding box and one invisible window. These are deliberate simplifications: while invisible lights can contribute significant illumination, they are hard to infer using only indirect cues. We limit the expressivity of the representation to account for this ill-posedness and find it to be a good choice in practice[1]. When a scene has no invisible light sources, their predicted intensity is close to zero, as shown in Figure 8.3 and Figure 8.8. To learn a better separation of the contributions of visible and invisible light sources, we provide a mask $\mathbf{M} = \sum_{\mathtt{W}} \mathbf{M}_{\mathtt{W}} + \sum_{\mathtt{L}} \mathbf{M}_{\mathtt{L}}$ of all visible sources to the invisible light estimation networks:

$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{z}\} = \textbf{InvLampNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}),$$
$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{SG}_{\text{sun}}, \mathbf{SG}_{\text{sky}}, \mathbf{SG}_{\text{grd}}\} = \textbf{InvWinNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}).$$

8.4 Neural Rendering Framework

In order to achieve photorealistic editing of indoor lighting, we need a rendering framework that can handle complex light transport typical for indoor scenes, including sharp directional lighting, hard and soft shadows, global illumination and specular materials. While existing differentiable path tracers can handle all these effects, they are computationally expensive. Even more importantly, they require the full reconstruction of material, geometry and lighting of the entire indoor scene, including its invisible parts.

To address these limitations, we introduce a neural rendering framework that combines the advantages of physically-based rendering and learning-based rendering, which works with

[1]The real scene in Figure8.1 has four invisible lamps and the last real scene in Figure 8.6 has 2 invisible lamps. In both cases, our method achieves reasonable approximation with one single invisible lamp.

our novel light source representations, does not require reconstruction of invisible scene surfaces, achieves high performance, and supports differentiability. Our framework, illustrated in Figure 8.3 (right), has 3 modules:

1. A physically-based direct shading module that computes the direct irradiance at a surface point from each light source through Monte Carlo sampling, with shadows rendered by casting rays against a mesh constructed from depth information.
2. An indirect shading module that predicts indirect shading (global illumination) from direct shading.
3. A per-pixel lighting reconstruction module that turns the shading, materials and geometry predictions into per-pixel lighting (i.e. a spatially-varying grid of incoming radiance environment maps) which can be used to render specular materials and insert objects.

Thus direct diffuse shading and occlusion is computed using classical techniques, while global illumination is added through neural rendering. The reason for this separation is that in the absence of full scene reconstruction (i.e. invisible parts), global illumination can only be computed heuristically, which is a task suited for neural networks. Conversely, direct illumination and non-local shadowing can be efficiently computed by standard techniques, but remain tricky for neural methods.

8.4.1 Direct Shading Rendering Module

We use inspiration from physically-based rendering [159] to sample the surface of each light source and connect those samples to the scene points. Formally, let $\mathbf{p}$ be a shading point and $\mathbf{q}$ be a point uniformly sampled on the light surface, with $\mathbf{p}\rightarrow\mathbf{q}$ the unit vector from $\mathbf{p}$ to $\mathbf{q}$. The direct shading $\mathbf{H}_j$ caused by light source j is computed as:

$$\mathbf{H}_j(\mathbf{p}) = \frac{\text{area}(j)}{N_j} \sum_{\mathbf{q}} \frac{\mathbf{L}_j(\mathbf{q}\rightarrow\mathbf{p}) \max(\cos\theta_{\mathbf{p}} \cos\theta_{\mathbf{q}}, 0)}{||\mathbf{q}-\mathbf{p}||_2^2}, \tag{8.2}$$

Table 8.2. Shadow rendering error with or w/o network inpainting.

	Ray traced	Ours
L_2	0.011	0.005

where $\cos\theta_{\mathbf{p}} = \mathbf{p}{\rightarrow}\mathbf{q} \cdot \mathbf{N}(\mathbf{p})$, $\cos\theta_{\mathbf{q}} = \mathbf{q}{\rightarrow}\mathbf{p} \cdot \mathbf{N}(\mathbf{q})$ and N_j is the number of samples for light source j. While our Monte Carlo estimation in (8.2) converges fast for lamps, it is not optimal for high-frequency directional sunlight coming through windows, since only when $\mathbf{q}{\rightarrow}\mathbf{p}$ aligns with the sun direction, will the $\mathbf{L}(\mathbf{q}{\rightarrow}\mathbf{p})$ return a significant contribution. To tackle this issue, with $\mathbf{Pr}(\mathbf{l})$ the probability of sampling direction ω from $\mathbf{SG}_{sun}$, we also generate samples according to the angular distribution of $\mathbf{SG}_{\text{sun}}$:

$$\mathbf{H}_j(\mathbf{p}) = \sum_{\mathbf{l}} \frac{\mathbf{L}_j(\omega)\mathbf{I}_j(\omega)\max(\cos\theta_{\mathbf{p}}, 0)}{N_j \mathbf{Pr}(\mathbf{l})}, \tag{8.3}$$

where $\mathbf{I}_j(\omega)$ is an indicator function to detect if ray ω starting from $\mathbf{p}$ can hit the window plane. Note that both (8.2) and (8.3) are unbiased but with different variances, which we combine with multiple importance sampling (MIS) [205]. Figure 8.4 compares the direct shading of a window, where we observe that our MIS method can render high-quality direct shading with fewer samples.

8.4.2 Depth-based Shadow Rendering Module

Recall that in the above shading computation, $\mathbf{H}_j, j \in \{\texttt{W}\} \cup \{\texttt{L}\}$ does not consider visibility and therefore cannot handle shadows. We could check visibility by ray-tracing during the Monte Carlo sampling above, but this causes artifacts due to incomplete geometry, as shown in Figure 8.6. We instead design a depth-based shadow rendering framework that combines Monte Carlo ray tracing with deep network inpainting and denoising. Note that our shadow modules are not differentiable, as this is not necessary for our application: we train our network on a synthetic dataset, where it is provided with the ground-truth supervision of direct shading without the shadow effects, so back-propagation of error through the shadow renderer is not used

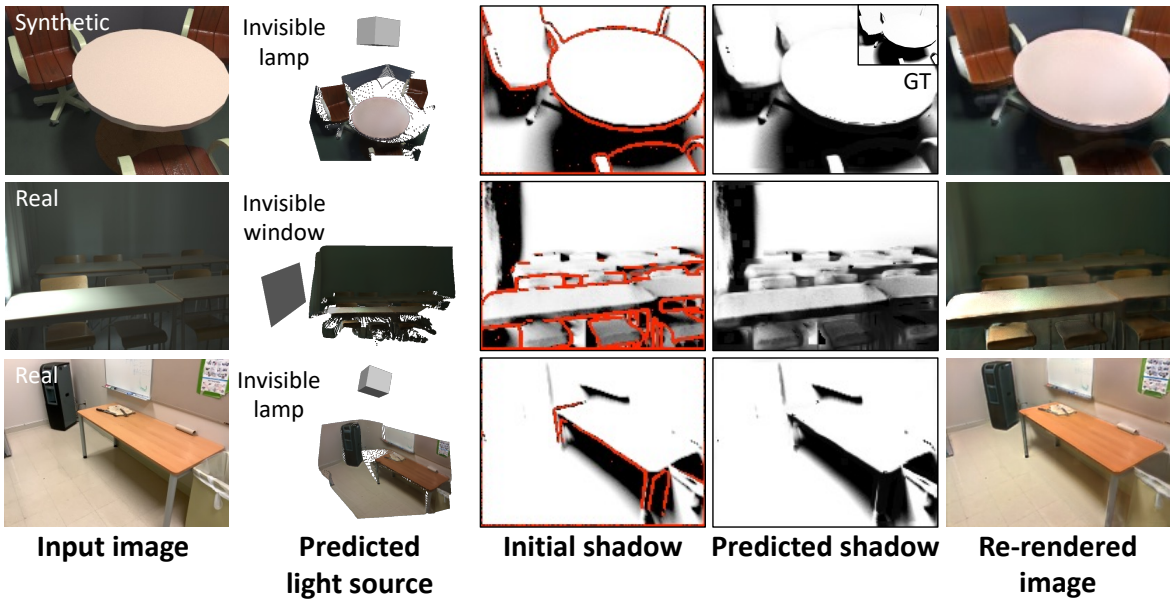


Figure 8.6. Our depth-based shadow rendering results. Direct rendering shadows from depth with ray tracing leads to boundary artifacts as shown in the red color in the third column. Our trained deep network successfully fixes the boundary issues and achieve high-quality shadows for both real and synthetic scenes, leading to re-rendered images that closely match the input images.

during training.

Our approach first creates a mesh from the depth map, and then uses a GPU-based ray-tracer to cast shadow rays from surfaces to light sources. To address the boundary artifacts, we first modify the renderer to detect the occlusion boundaries, then train a CNN to fill in the shadow at these regions. This hybrid approach outperforms both pure ray-tracing and a CNN trained to clean up the entire ray-traced shadow image. Formally, let $\mathbf{S}^{\mathbf{Init}}$ be the initial shadow image rendered from depth map $\mathbf{D}$ and let $\mathbf{M}^{\mathbf{S}}$ be the mask for occlusion boundaries. We have:

$$\mathbf{S} = \mathbf{M}^{\mathbf{S}} \cdot \mathbf{DShdNet}(\mathbf{S}^{\mathbf{Init}}, \mathbf{D}, \mathbf{N}) + (\mathbf{1} - \mathbf{M}^{\mathbf{S}}) \cdot \mathbf{S}^{\mathbf{Init}}. \tag{8.4}$$

The total direct shading from all sources is $\mathbf{H_d} = \sum_j \mathbf{H}_j \mathbf{S}_j$. As seen in Figure 8.1, 8.6 and 8.7, our framework can render higher quality soft and hard shadows that are closer to the ground-truths compared to a standard ray tracer. Table 8.2 shows that our CNN reduces the shadow error by

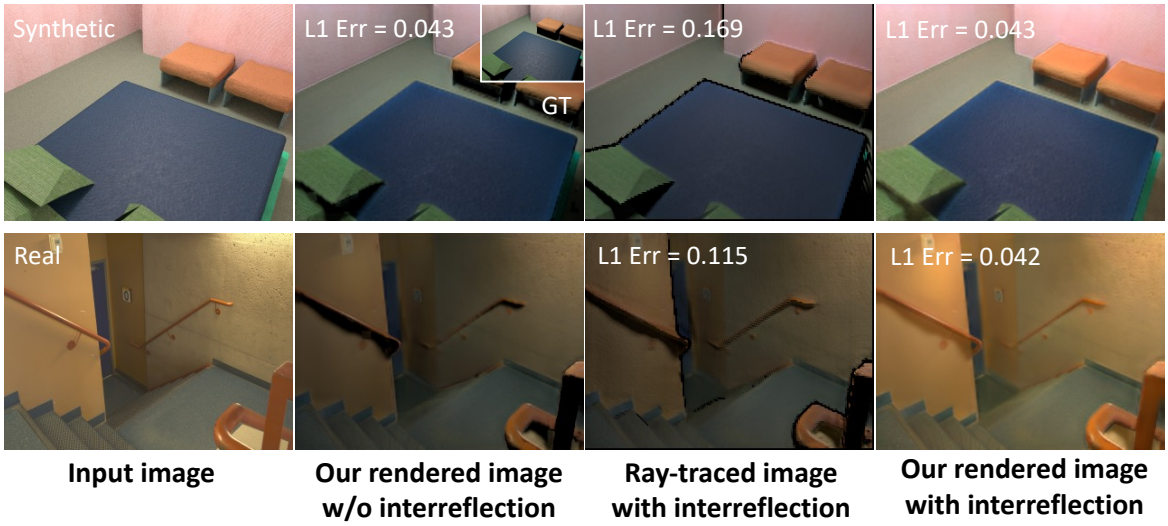


Figure 8.7. Comparisons of images directly rendered from standard path tracer using partial observation from a single view and from our neural rendering framework. We show quantitatively and qualitatively that our renderer renders both direct and indirect illumination accurately, while a path tracer cannot model indirect illumination without complete reconstruction of scene geometry and materials and has artifacts near the occlusion boundaries.

more than 50%.

8.4.3 Indirect Shading Prediction

To render indirect shading with a standard physically-based renderer, we would need to reconstruct invisible geometry and materials, which is challenging. Instead, we train a 2D CNN to predict indirect shading in screen space. A similar idea has been adopted by recent work [222]. We use a network with large receptive field covering the entire image to model non-local inter-reflections. Our indirect shading is $\mathbf{H_{Ind}} = \mathbf{IndirectNet}(\mathbf{H_d}, \mathbf{D}, \mathbf{N}, \mathbf{A})$, which is added to the direct shading for the final shading prediction. In Figure 8.7, we compare the indirect illumination rendered by our network and by a standard path tracer by first building a mesh from the depth map and then texturing the mesh with predicted materials. Quantitative and qualitative results on real and synthetic examples show that our neural rendering layer renders both direct and indirect illumination accurately, while a standard path tracer cannot handle indirect illumination with partial geometry, leading to an image with similar intensity as one

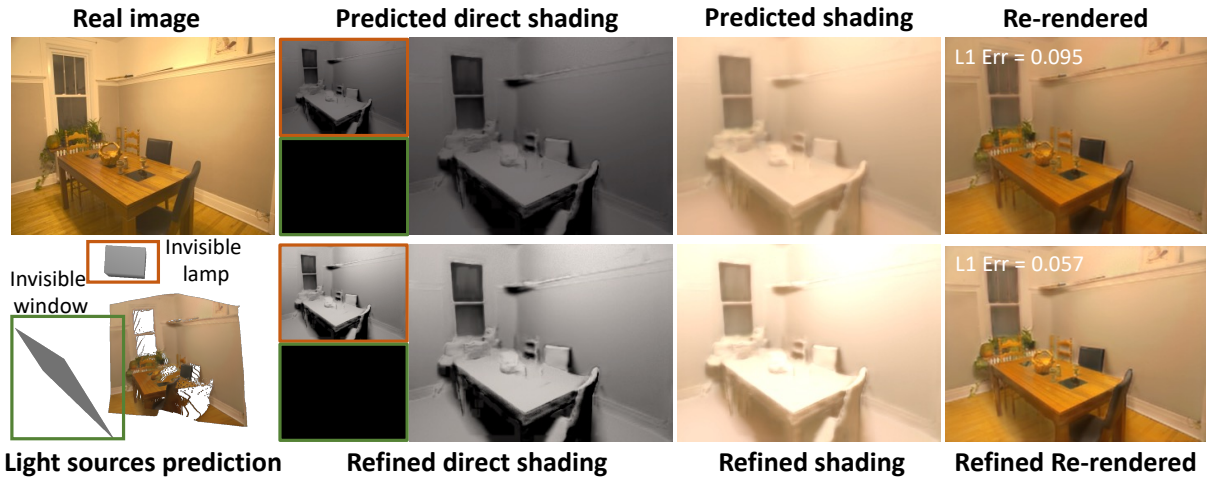


Figure 8.8. Comparisons of light source prediction and rendering before and after the optimization on a real image. Our differentiable neural rendering framework allows us to use the rendering loss to learn and refine light source intensity, angular distribution and direction, leading to more realistic rendering of the scene.

with direct illumination only.

8.4.4 Predicting Lighting From Shading

The above framework cannot yet handle specular reflectance, which motivates us to add another network to infer spatially and directionally varying incoming lighting $\mathbf{L}$, taking the above shading (irradiance) $\mathbf{H}$ as input. We follow [115] to predict a grid of environment maps. We use a similar network architecture but replace the input image $\mathbf{I}$ with the shading $\mathbf{H}$ so that the predicted *local* lighting is a function of our lighting representation: $\mathbf{L} = \mathbf{LightNet}(\mathbf{H}, \mathbf{M}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D})$. The resulting incoming radiance field $\mathbf{L}$ can be used to render specular materials, as shown in Sec. 8.6.

8.5 Implementation Details

Dataset

We train our framework on OpenRooms proposed in Chapter 6, which is a large-scale synthetic indoor dataset for inverse rendering. It provides all ground truths for all our outputs, including some unique ones such as light source geometry, per-light source shadings (with and w/o occlusion) and per-light source shadows. Thus, it allows us to train each module separately,

which significantly simplifies the training.

Optimized light source parameters

We augment the OpenRooms dataset with optimized light source parameters $\mathbf{SG}_{\text{sun}}$, $\mathbf{SG}_{\text{sky}}$ and $\mathbf{SG}_{\text{grd}}$ for windows, leading to sharper and more interpretable predictions. To compute those, we minimize the L_1 difference between the rendered direct shading without occlusion $\mathbf{H}_j$, $j \in \{\texttt{W}\}$ and its corresponding ground truth, through our differentiable Monte Carlo rendering module (Sec 8.4.1). The optimized direct shading is seen in Figure 8.4 to closely match the ground truth.

Losses

To train **MNet**, we use L_2 loss on the albedo, normal and roughness. The loss function for light source prediction is the sum of a rendering loss ($\mathscr{L}_{ren}$), a geometry loss ($\mathscr{L}_{geo}$), and a light source loss ($\mathscr{L}_{src}$). For $\mathscr{L}_{ren}$, we define it to be the L_1 distance between the rendered direct shading $\mathbf{H}_j$ and its ground-truth direct shading, both without shadows applied so that we can avoid the relatively time-consuming shadow rendering during training (Table 8.3). For $\mathscr{L}_{geo}$, we uniformly sample sets of points $\{\mathbf{q}\}$ from the ground-truth and predicted light source geometry to compute their L_1 Chamfer distances and add an L_1 loss for the area of the light sources to encourage sharper lighting. Finally, for $\mathscr{L}_{src}$, we use L_2 loss for direction $\mathbf{d}$, $\log L_2$ loss for intensity $\mathbf{w}$ and bandwidth λ. To train the shadow network, we use scale-invariant gradient loss proposed in [150] and find that it leads to many fewer artifacts compared to a simple L_2 loss. We supervise indirect shading with L_1 loss and per-pixel lighting with rendering loss and $\log L_2$ loss similar to [115].

Training

We use Adam [103] with learning rate 10^{-4} and β (0.9, 0.999). We first train the **MNet**, then we fix it and use its predictions as inputs to train the light source prediction networks **InvLampNet**, **InvWinNet**, **VisLampNet** and **VisWinNet** separately. We then train rendering

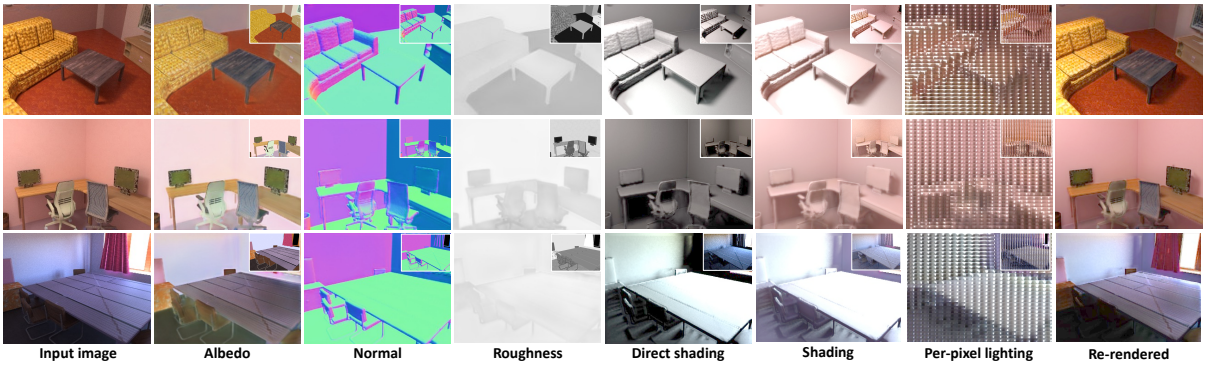


Figure 8.9. Material predictions and neural rendering results on the OpenRooms synthetic testing set. The ground-truths are shown in the insets on the up-right corner. We observe that our method can render high-quality direct shading, shading, per-pixel environment map and final image from our light source and material predictions, with non-local shadows and interreflection being correctly modeled.

Table 8.3. Inference time of each step of our framework.

Material	Light source	Direct shading	Shadow	Indirect shading	Per-pixel lighting	Total
299ms	19.7ms	595ms	1309ms	19.1ms	19.35ms	2.26s

modules independently by providing them with ground-truth $\mathbf{H_d}$ and $\mathbf{S}$ whenever they are required as inputs.

Inference time

The inference time for the network to process one image is summarized in Table 8.3. The most time consuming stage is to render shadows from depth using path tracing. Note that while our framework handles many complex light transport effects, the total time for it to reconstruct and re-render an indoor scene is less than 3 s.

Refinement

While so far our framework can achieve high-quality light source prediction and indoor lighting editing in many cases, our differentiable neural rendering framework enables us to further refine the light source parameters by minimizing the rendering loss between the rendered image and the input image, leading to more robust and more realistic rendering. Figure 8.8 shows an example where we correct the intensity of an invisible lamp with our rendering loss-based

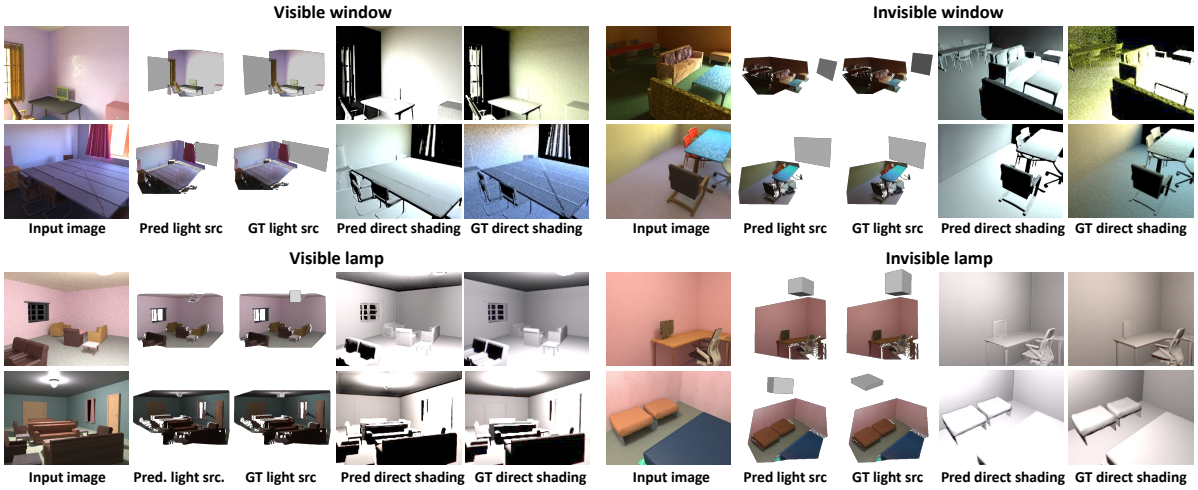


Figure 8.10. Light source prediction results on the synthetic dataset for various types of light sources. We visualize the predicted and ground-truth light source geometry and their rendered direct shading $\mathbf{H}_j$ without occlusion. Our method can recover both the geometry and radiance for the four types of light sources reasonably well, leading to rendered direct shading $\mathbf{H}_j$ similar to the ground-truths.

refinement. Note that this is an extremely ill-posed problem. A good initialization from our light source prediction networks is essential for the refinement to achieve good results. We only apply the refinement to real images shown in the paper, not to the synthetic images.

8.6 Experiments

We first present extensive quantitative and qualitative evaluations on the synthetic OpenRooms dataset [119]. Then, we focus on qualitative demonstrations of light source predictions and various scene editing applications, especially light editing, on real indoor scene images. For synthetic examples, we use ground-truth depths and segmentation masks. For real examples, we generate depth predictions using DPT [169] and manually label the light source segmentation masks.

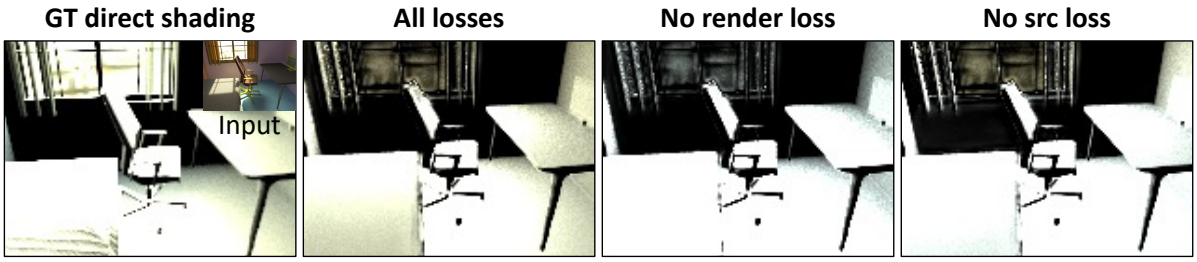


Figure 8.11. Ablation studies on different loss combinations for window light source prediction. Our network trained with both rendering loss $\mathscr{L}^j_{ren}$ and light source loss $\mathscr{L}^j_{src}$ predicts the most accurate radiance, with both high-frequency directional lighting and ambient lighting closely matching the ground-truth.

Table 8.4. Material predictions on the OpenRooms testing set. We report L_2 error of our material predictions.

	Albedo 10^{-2}	Normal 10^{-2}	Roughness 10^{-2}
Ours	1.81	1.39	6.22
Chapter 7	-	4.51	6.59

8.6.1 Experiments on Synthetic Images

Material prediction

Our material prediction errors are summarized in Table 8.4. Note that unlike in Chapter 7, which first uses scale-invariant loss for albedo prediction and adopts a linear regression to solve the scale ambiguity, we use the absolute loss for both diffuse albedo and light intensity prediction. The reason is that our method needs to recover the radiance of multiple light sources in the scene and it is difficult to recover consistent intensities across multiple light sources through simple linear regression. Table 8.4 compares our material prediction with [119]. Since we use ground-truth depth as an input, our normal prediction is much more accurate compared to [119]. The roughness accuracies are very similar. In Figure 8.9, we present material predictions on our synthetic testing set. On synthetic data, we show that both our diffuse albedo and roughness predictions are very similar to the ground-truths.

Table 8.5. Light source prediction losses on our synthetic testing set. We report L_1 Chamfer distance for geometry and L_1 error of direct shading $\mathbf{H}_j$ for rendering.

	Geometry	Rendering
	Chamfer (m) $\mathbf{Cham}(\mathbf{q}_j, \hat{\mathbf{q}}_j)$	Direct shading $\mathbf{H}_j$ $\mathbf{H}_j$
Vis. lamp	0.351	0.318
Vis. window	0.857	0.849
Inv. lamp	1.582	0.289
Inv. window	5.500	0.312

Table 8.6. Ablation studies on window light source prediction. We report L_1 loss for direct shading $\mathbf{H}_j$, L_2 loss for direction $\mathbf{d}$ and $\log L_2$ loss for intensity $\mathbf{w}$ and λ.

Visible window	Rendering	Light source		
	Direct $\mathbf{H}_j$	Intensity $\mathbf{w}$	Direction $\mathbf{d}$	Bandwidth λ
w/o $\mathscr{L}^j_{ren}$	1.276	**7.972**	0.386	**4.369**
w/o $\mathscr{L}^j_{src}$	0.859	17.73	0.503	7.492
All	**0.849**	10.28	**0.369**	4.419
Invisible window	Rendering	Light source		
	Direct $\mathbf{H}_j$	Intensity $\mathbf{w}$	Direction $\mathbf{d}$	Bandwidth λ
w/o $\mathscr{L}^j_{ren}$	1.786	**10.817**	0.545	**4.770**
w/o $\mathscr{L}^j_{src}$	0.334	44.04	1.432	70.48
All	**0.312**	18.15	**0.536**	8.168

Light source prediction

Figure 8.10 shows qualitative results of our light source predictions on the synthetic testing set. For each of the four types of light sources, we pick up two examples from the testing set to visualize their predicted and ground-truth geometry and their rendered direct shading $\mathbf{H}_j$ without occlusion. We observe that our method can recover both the geometry and radiance for all 4 types of light sources reasonably well, which enables us to render their direct shading quite close to the ground-truths. We notice that the major errors in our rendered direct shading are caused by global shifts of colors and intensities, while the locations of highlight are usually correct. This is reasonable given the scale ambiguity between albedo and lighting. However, even with these global shifts in albedo and light intensity predictions, as shown in both Figure

Table 8.7. Quantitative errors for our neural rendering framework. We report L_1 loss for direct shading $\mathbf{H_d}$ and shading $\mathbf{H}$. We report $\log L_2$ loss for per-pixel lighting $\mathbf{L}$.

Direct shading $\mathbf{H_d}$	Shading $\mathbf{H}$	Per-pixel lighting $\mathbf{L}$
0.256	0.317	0.091

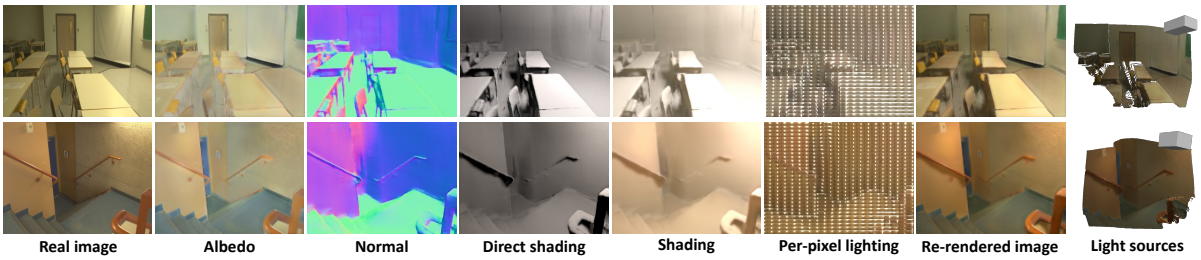


Figure 8.12. Our reflectance, geometry, lighting and rendering predictions two real examples. We observe that even for invisible light sources, our framework accurately reconstructs their geometry and intensities, which enables realistic rendering of the scene irradiance, shadows, interreflection and per-pixel lighting.

8.9 and Figure 8.12, our final rendered images can closely match the input images, for both real and synthetic examples. Table 8.5 summarizes the quantitative errors for all types of light sources. We see that the errors for windows are always larger than those of lamps because the outdoor lighting coming through windows is much more complicated compared to indoor area lighting. We also observe that the direct shading errors for invisible light sources are lower. This is because their overall contributions are usually lower since some of them can be far away from the camera location.

Table 8.6 and Figure 8.11 verifies the effectiveness of our loss functions for window light source prediction. We observe that while training with light source loss $\mathscr{L}_{src}^{j}$ can lead to the prediction closest to our optimized ground-truth light source parameters, the rendering error is significantly higher because it is difficult to find the best balance across different parameters that can minimize the rendering error. Training with $\mathscr{L}_{ren}^{j}$ alone leads to reasonable direct shading prediction. However, the light source parameters are less interpretable, as shown in Table 8.6 and the rendered direct shading tends to be oversmoothed, as shown in Fig 8.11. Combining the two losses together, on the contrary, allows us to render direct shading closer to the ground-truth, with

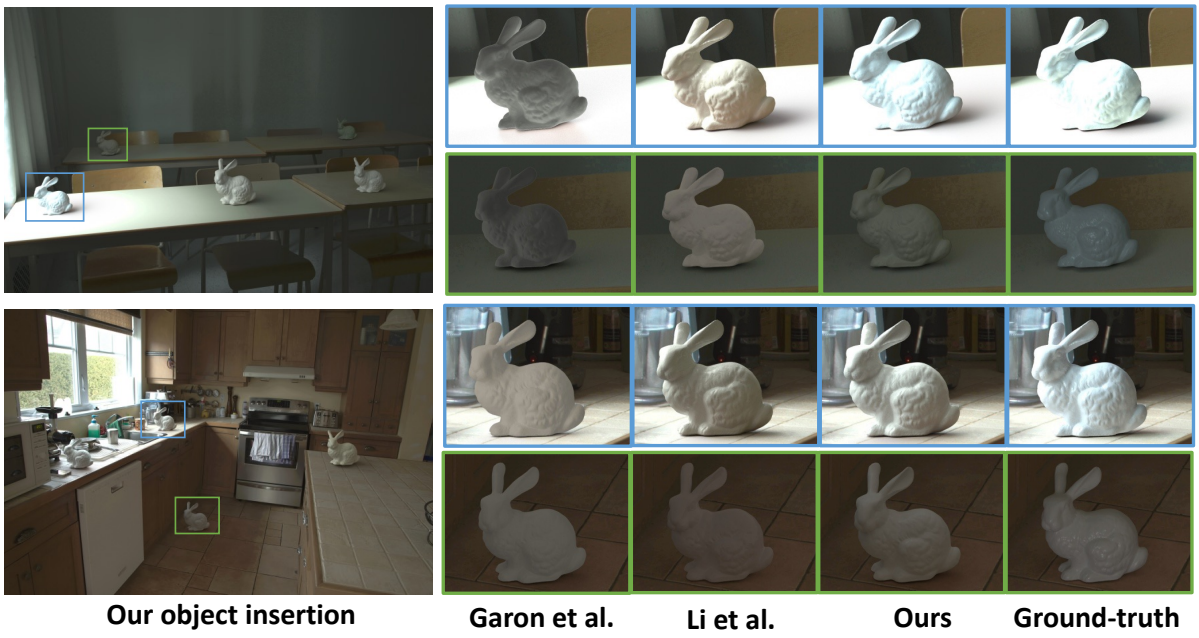


Figure 8.13. Comparions with prior lighting estimation methods on Garon et al. [63] dataset for object insertion on the surface of the scenes. Our method achieves results with similar quality compared to the state-of-the-arts. Specifically, our method accurately reconstructs the complex lighting from windows and therefore can render more realistic highlights and shadows.

Table 8.8. User study on Garon et al. dataset. We require users to compare our object insertion results with prior results and report the percentage of users who believes ours are better.

Gardner et al. [61]	Garon et al. [63]	Chapter 7
72.4%	69.2%	52.0%

high-frequency lighting being correctly modeled, as shown in both Table 8.6 and Figure 8.11.

Neural rendering

Quantitative and qualitative results of our neural rendering framework on the synthetic testing set are summarized in Table 8.7 and Figure 8.9 respectively. We see that from a single LDR image, our method first predicts both the geometry and radiance of the light sources accurately, which enables us to render direct shading with both shadow and intensity very similar to the ground-truths. Our shadow prediction network combined with Monte-Carlo ray tracing allows for rendering distant shadows from a single depth map without boundary artifacts, and our indirect shading prediction network models non-local interreflection from only partial observation

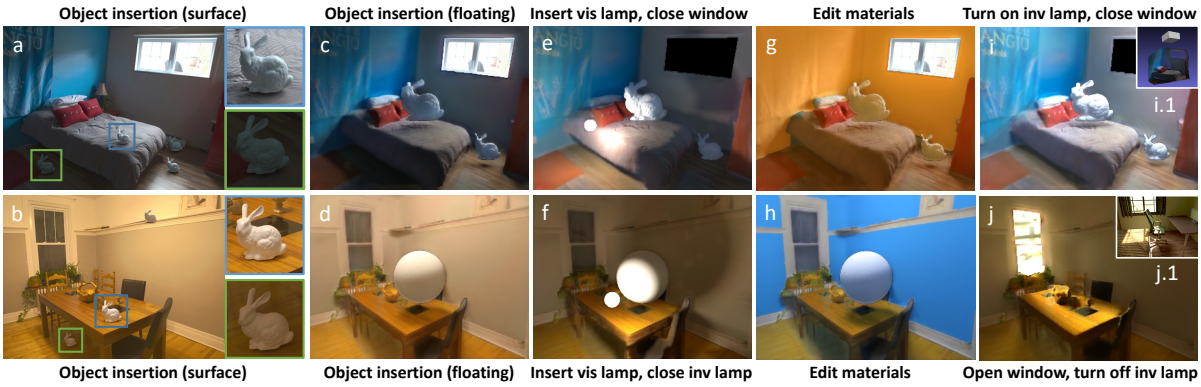


Figure 8.14. Various editing applications demonstrated on 2 real examples. In addition to high-quality object insertion results (a and b), our light source representations combined with our neural rendering framework allows us to edit geometry, material and lighting of indoor scenes with non-local effects being effectively modeled. This includes distant shadows projected to the bed or table (c and d) or to the entire room when the object blocks the light source (e and f), changing color of walls that causes non-local color bleeding (g and h) and adding virtual light sources into the scene (e, f, i, j), including turning on a lamp or opening a virtual window.

of geometry and materials. All this combined together leads to accurate reconstruction of shading, per-pixel lighting and re-rendered images that closely matches the input images.

8.6.2 Experiment on Real Images

Light source predictions and neural rendering

Figure 8.12 shows a complete set of our material predictions, light source predictions and neural rendering results on two real examples. Even though we do not have ground-truths, we observe that the light source position, the highlight in the direct shading and the color bleeding in shading caused by interreflection from walls are all visually consistent. The re-rendered images with appearance closely matching the input images further demonstrate that our framework can correctly reconstruct both radiance and geometry of light sources, generalizes well to real examples and re-render the image by explicitly considering various challenging light transport effects, which allows us to achieve much higher quality compared to a standard physically-based renderer under incomplete observation.

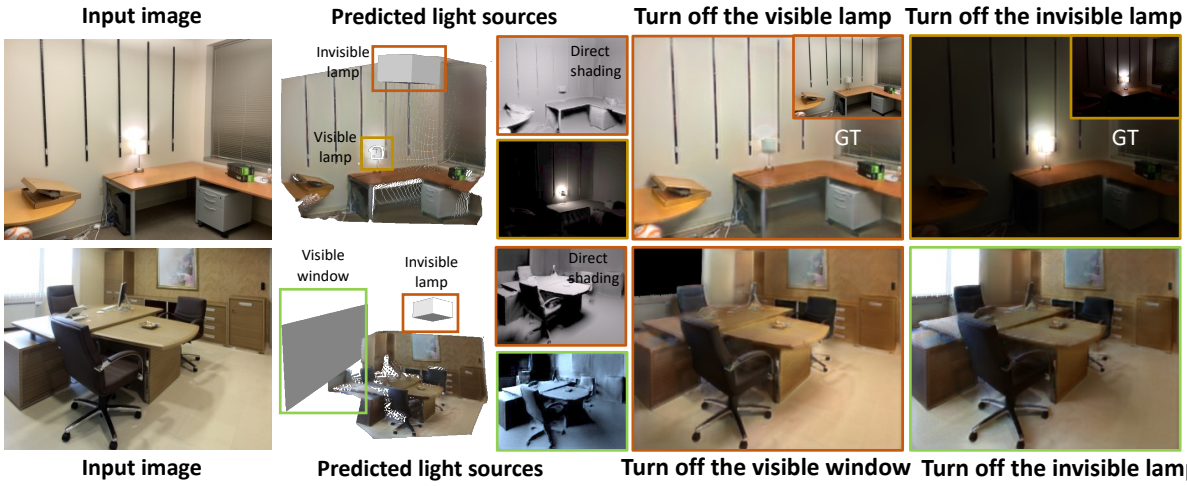


Figure 8.15. Our accurate reconstruction of visible/invisible light sources allows us to separate their contributions and turn them on and off. We show two examples here. For the first example captured by ourselves, we observe that our editing results closely match the ground-truth insets. For the second example from the internet, even though we do not have ground-truths, we see that our editing results are realistic and visually consistent.

Comparisons with prior works

We compare with prior state-of-the-art lighting estimation methods on object insertion using the Garon et al. dataset, which is a real dataset containing ground-truth HDR spatially-varying lighting annotation. Even though we are solving a harder problem, both qualitative and quantitative results in Figure 8.13 and Table 8.8 show that our method achieves performance comparable to the prior state-of-the-arts which only handle local editing of the scene. Our per-pixel lighting prediction can be used to render specular objects realistically, with highlights, shadows and spatial consistency being correctly modeled as shown in Figure 8.13 and 8.14. Specifically, our 3 SG sunlight representation and MIS based rendering layer allow us to better handle high-frequency, complex sunlight coming from the window, leading to rendering results closer to the ground-truths, as presented in Figure 8.13.

Novel scene editing applications

In addition to high-quality object insertion with local highlights and shadows, the true advantage of our framework is its ability to handle non-local effects in scene editing application,

which is only made possible by our accurate reconstruction of indoor light sources and high-quality neural rendering that models multiple complex light transport effects, such as hard/soft shadows, interreflection and directional lighting. These non-local effects include distant shadows and highlights, which is shown in (c and d) of Figure 8.14 where the inserted bunnies block the sunlight coming from the visible window and the sphere blocks the light from the invisible lamp, projecting shadows to the bed, floor and table respectively. This is further demonstrated in (e and f), where the inserted virtual lamp causes highlights on the surface of nearby geometry and causes shadows that cover the whole wall behind the inserted virtual bunny and sphere. Moreover, our framework also allows non-local interreflection to be accurately modified. As shown in (g and h), as we change the color of walls to orange and blue, our indirect shading network paints the inserted white objects with correct color bleeding. In (i and j), we further demonstrate our framework's ability to turn on an invisible lamp or open a virtual window. Note that in j, we use the 3 SG approximation of the environment map shown in j.1. Our representation combined with our neural renderer can render realistic sharp directional sunlight.

Our accurate reconstruction of indoor light sources further allows us to separate their contributions. As shown in both Figure 8.1 and 8.15, our framework allows us to turn off both visible and invisible, windows and lamps in the scene, with realistic changed appearance that are very similar to the ground-truths.

8.7 Conclusions

We presented a method that enables full indoor scene relighting and other editing operations from a single LDR image with its predicted depth and light source segmentation mask. A key innovation in our solution is our lighting representation; we estimate multiple global 3D parametric lights (lamps and windows), both visible and invisible. A second important component is our hybrid renderer, capable of producing high-quality images from our scene representation using a combination of Monte Carlo and neural techniques. We demonstrated that

this careful combination of an editable lighting representation and neural rendering can handle challenging scene editing applications including object insertion, material editing, light insertion and light editing, with realistic global illumination effects.

Chapter 8 is based on the material currently under submission ("Physically-Based Editing of Indoor Scene Lighting from a Single Image", Zhengqin Li1 Jia Shi, Sai Bi, Rui Zhu, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, Manmohan Chandraker). The dissertation author was the primary investigators and author of this paper.

Chapter 9

Conclusion and Future Work

This dissertation has inspired significant related research on scene reconstruction and inverse rendering, including measurement-based [127], model-based [125, 151] and learning-based methods [182, 197, 224], spanning material estimation [60], joint shape and material estimation [28, 177], lighting estimation [193] and holistic indoor scene understanding [208, 181]. Also notable are lines of work that build upon this progress to consider differentiable rendering for inverse rendering problems [27, 24, 23, 125, 151]. In this chapter, we will first describe a few other directions in which future works might extend this thesis (Section 9.1), before concluding with perspectives gained through our approach of physically-motivated learning for reconstruction and editing in complex scenes (Section 9.2).

9.1 Future Work

Consistency in scene reconstruction

This thesis has largely considered the most challenging input for scene reconstruction problems, namely a single image captured in an unconstrained environments. Having multiple images of the scene as input makes the problem better-posed, but also requires consistency of predictions across those inputs. This includes multiview consistency – given multiple images of the same scene, the reconstructed intrinsic components at the overlapping regions should be the same, as well as temporal consistency – given a video taken across a time span, the ideal predictions should be stable between frames without flickering artifacts. A classical way to

achieve consistency is to formulate it as an optimization problem [126], where the consistency constraint can be used to compute a loss term. Our differentiable rendering module allows back-propagating gradients to the network for optimization purposes, which may offer a powerful and effective tool for achieving consistent scene reconstruction.

More complex appearances

In this thesis, we have largely assumed that material appearance can be modeled by an opaque BRDF where rays only get reflected when they hit surfaces, or by a perfectly transparent material. This does not allow modeling translucent objects such as jade, where a ray can travel into the surface while experiencing an infinite number of scattering and absorption events before emerging. The appearance of such materials must be modeled by a bidirectional scattering distribution function (BSDF) and to render such appearance, we need volume rendering, which is computationally expensive. Similarly, volumetric scattering to represent participating media such as fog or mist is not handled by this dissertation. Measurement-based methods develop systems with controllable lighting to capture the appearance of translucent objects [67]. Recently, Che et al. [39] make the first attempt to build a deep learning-based framework with a physically-based Monte Carlo rendering layer for recovering subsurface scattering parameters. However, there is still large space for improvement to make translucent material reconstruction more robust and practical.

Neural scene representations

Recently, neural implicit representations have become popular in computer vision and graphics, for view synthesis [137], geometry reconstruction [206], pose estimation [228] and appearance modeling [22]. Instead of explicitly representing geometry as points and faces in 3D space, they use a multi-layer perceptron that outputs an RGB color and a density value to indicate whether a 3D point in space is occupied [137]. A significant advantage of this representation is infinite resolution – one may densely sample points in a region without worrying about the memory consumption, which is not possible for explicit representations such as

meshes, point clouds or volumes. In addition, neural scene representations can be combined with differentiable volume rendering, which yields a powerful tool to directly optimize scene appearance with occlusion and reflection being easily handled. However, most recent inverse rendering frameworks based on this representation are still computationally expensive and face difficulties handling complex light transports such as multi-bounce interreflections in large-scale scenes. The combination of such neural scene representations with the advantages of generalization ability, scalability and high-quality intuitive editing enabled by our physically-motivated design principles promises to be a highly impactful direction of future research in computer vision, computer graphics and machine learning.

9.2 Perspectives

This thesis has considered a fundamental challenge in computer vision and computer graphics: recovering geometry, lighting, and material of the whole scene from a single or a few images captured under constrained conditions, despite the presence of complex light transport. It builds upon and serves as a counterpoint to multiple distinguished lines of work in those fields, which have sought to measure various intrinsic scene properties, or model the underlying physical basis of image formation. While classical methods provide valuable insights into what is needed to succeed at scene reconstruction, they have not lent themselves to robust and practical optimization for the highly ill-posed nature of the problem. Data-driven priors learned by deep neural networks, thus, emerged as viable means of regularization.

But modern deep networks are extremely data-hungry, while the challenges of inverse rendering necessitate reasoning about diverse scene appearances and light transport effects for which ground truth supervision might be difficult or well nigh impossible to obtain. The philosophy espoused by this dissertation is that the rich history of insights into the geometric and physical processes that govern image formation and constrain light transport can be valuable in establishing a learning paradigm that overcomes the fundamental limitations of both physically-

based and data-driven approaches. Such a paradigm must encompass all aspects of a learning problem, namely, architecture, representation and data. This dissertation has systematically addressed those aspects to solve problems in scene reconstruction that were hitherto considered intractable.

Physically-motivated neural architectures

A key advance achieved by this dissertation is the development of differentiable in-network rendering modules that rely on physically-based models of image formation to accurately model non-local, complex light transport effects such as specularity, shadows and interreflections. This overcomes the significant challenge of generalization by imbibing inductive biases of the physics of light transport, with the additional benefit of network outputs that are interpretable and physically meaningful. While the former is an important need for the solution to be tractable, the latter is equally important for scene understanding outputs to be useful in a downstream applications such as editing, AR or 3D modeling where users must interact with images or 3D content in tangible and intuitive ways. While the ever-increasing volume of data and the power of even newer architectures like transformers will continue to enhance the capabilities of black-box machine learning, we believe that the advantages of generalizability and intuitive interpretability achieved by the proposed physically-motivated learning paradigm transcend advances in base neural architectures.

Physically-based representations

Representations for shape amenable to deep learning have seen tremendous progress over the years, but inverse rendering must deal with a next level of challenges due to complex material and lighting. Early works in computer graphics and vision have established models for material behavior that are highly descriptive yet compact enough for real-time rendering. Such material representations have proven to be influential in our physically-motivated learning where in-network rendering plays an important role. Similarly, lighting representations have a long history in computer graphics, where they have been studied to understand tradeoffs with respect

to representation power, storage and computational expense. We build upon this history, but go beyond it to devise representations that are efficient for the particular nature of computations that dominate learning with deep neural networks. In some cases, such as light sources in indoor scenes, the dual needs of extremely accurate localization and modeling of long-range effects across distant parts of the scene necessitates carefully-designed new representations. The ones we develop are the first of their kind to accurately model geometry and color, while being able to leverage a rich set of classical approaches such as Monte Carlo ray tracing, shadow denoising and modeling of indirect irradiance.

Open frameworks for physically-motivated data and environments

An equally important perspective gained from this dissertation is on the importance of data that is not just relevant, but accessible. The nature of inverse rendering problem requires diverse ground truth, while our physically-based modeling benefits from supervision for spatially-varying properties or complex light paths that are difficult to measure in real scenes. This points to photorealistic synthetic datasets that are designed with physically-based rendering to aid the proposed learning paradigm. However, we quickly realized that data of this nature is extremely difficult to procure and often of a proprietary nature that only a few research groups can afford. This led to the development of OpenRooms, which aims to democratize data generation and curation for inverse rendering to the extent that anyone with a conventional scanner or camera can create photorealistic datasets of their own. We believe this will be a significant step in making augmented reality truly mainstream, where consumers are not just users of proprietary content, but freely generating photorealistic and interactive content of their own.

Physically-based applications

The above advances herald a new era for scene reconstruction where shape, material and lighting have been estimated to a high degree of accuracy despite partial observations of complex light transport. Our frameworks outperform traditional methods significantly for capturing realistic SVBRDF and lighting from minimal inputs captured in the wild, while accurately

modeling various challenging visual effects such as color bleeding, hard and soft shadows, global illumination and refraction. This will lead to intuitive applications where images can be edited and augmented for downstream applications with unprecedented photorealism, using minimal acquisition effort. This thesis showcases several such applications, including virtual object insertion with consistent specular highlights and shadows, transparent shape relighting, novel view synthesis and relighting for specular objects, changing wall colors with realistic color bleeding on furniture, as well as photorealistic local and long-range effects from turning a lamp on or off, or opening a window to let in the sunlight to illuminate a complex indoor scene.

Bibliography

[1] Bullet Real-Time Physics Simulation. https://pybullet.org.

[2] Substance Share. https://share.substance3d.com/.

[3] Adobe stock. https://stock.adobe.com/3d-assets, 2017.

[4] Video. https://drive.google.com/file/d/1SV9pOERzrQk6RS6JoWGT0hm1sTPSYBt0/view?usp=sharing, 2022.

[5] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Trans. Graphics*, 35(4), 2016.

[6] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot svbrdf capture for stationary materials. *ACM Trans. Graphics*, 34(4), 2015.

[7] Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. Time-resolved 3D capture of non-stationary gas flows. *ACM ToG*, 27(5):132:1–132:9, December 2008.

[8] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

[9] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. SceneCAD: Predicting object alignments and layouts in RGB-D scans. In *Proc. ECCV*, 2020.

[10] Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. *arXiv preprint arXiv:1903.07145*, 2019.

[11] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2D-3D model alignment via surface normal prediction. In *CVPR*, 2016.

[12] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013.

[13] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *PAMI*, 37(8), 2015.

[14] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer, 2016.

[15] Harry G. Barrow and J. Martin Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, pages 3–26, 1978.

[16] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *PAMI*, 25(2), 2003.

[17] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):159, 2014.

[18] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014.

[19] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *CVPR*, 2015.

[20] Ben-Ezra and Nayar. What does motion reveal about transparency? In *ICCV*, pages 1025–1032 vol.2, 2003.

[21] Paul J. Besl and Neil D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[22] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020.

[23] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. *arXiv preprint arXiv:2007.09892*, 2020.

[24] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3D capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020.

[25] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, 1999.

[26] James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10), 1976.

[27] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. *arXiv preprint arXiv:2012.03918*, 2020.

[28] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik Lensch, and Jan Kautz. Two-shot spatially-varying brdf and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2020.

[29] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635. IEEE, 2014.

[30] Manmohan Chandraker, , and Ravi Ramamoorthi. What an image reveals about material reflectance. In *ICCV*, 2011.

[31] Manmohan Chandraker. On shape and material recovery from motion. In *ECCV*, 2014.

[32] Manmohan Chandraker. What camera motion reveals about shape with unknown BRDF. In *CVPR*, 2014.

[33] Manmohan Chandraker. The information available to a moving observer on shape with unknown, isotropic brdfs. *PAMI*, 38(7):1283–1297, 2016.

[34] Manmohan Chandraker, Dikpal Reddy, Yizhou Wang, and Ravi Ramamoorthi. What object motion reveals about shape with unknown BRDF and lighting. In *CVPR*, 2013.

[35] 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.

[36] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[37] Visesh Chari and Peter Sturm. A theory of refractive photo-light-path triangulation. In *CVPR*, pages 1438–1445, Washington, DC, USA, 2013. IEEE Computer Society.

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

[39] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020.

[40] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Learning transparent object matting. *IJCV*, 127(10):1527–1544, 2019.

[41] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2661–2670, 2019.

[42] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[43] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017.

[44] T. Chen, H. P. A. Lensch, C. Fuchs, and H. Seidel. Polarization and phase-shifting for 3D scanning of translucent objects. In *CVPR*, pages 1–8, June 2007.

[45] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, 2016.

[46] Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. Environment matting extensions: Towards higher accuracy and real-time capture. In *SIGGRAPH*, pages 121–130, 2000.

[47] Michael F Cohen and John R Wallace. *Radiosity and realistic image synthesis*. Elsevier, 1993.

[48] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[49] Robert L Cook and Kenneth E Torrance. A reflectance model for computer graphics. *ACM Trans. Graphics*, 1(1), 1982.

[50] Z. Cui, J. Gu, B. Shi, P. Tan, and J. Kautz. Polarimetric multi-view stereo. In *CVPR*, pages 369–378, July 2017.

[51] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[52] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, volume 98, pages 189–198, 1998.

[53] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, 2000.

[54] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Trans. Graph.*, 37(4), 2018.

[55] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (TOG)*, 37(4):128, 2018.

[56] Qi Duan, Jianfei Cai, and Jianmin Zheng. Compressive environment matting. *Vis. Comput.*, 31(12):1587–1600, December 2015.

[57] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[58] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[59] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3D-FUTURE: 3D furniture shape with texture. *arXiv preprint arXiv:2009.09633*, 2020.

[60] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Transactions on Graphics (TOG)*, 38(4):134, 2019.

[61] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Trans. Graphics*, 9(4), 2017.

[62] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *ICCV*, 2019.

[63] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2019.

[64] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *ICCV*, 2017.

[65] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Luc Van Gool, and Tinne Tuytelaars. Delight-net: Decomposing reflectance maps into specular materials and natural illumination. *arXiv preprint arXiv:1603.08240*, 2016.

[66] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Efstratios Gavves, Mario Fritz, Luc Van Gool, and Tinne Tuytelaars. Reflectance and natural illumination from single-material specular objects using deep learning. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1932–1947, 2017.

[67] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):1–13, 2013.

[68] Clement Godard, Peter Hedman, Wenbin Li, and Gabriel J Brostow. Multi-view reconstruction of highly specular surfaces in uncontrolled environments. In *3DV*, 2015.

[69] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *PAMI*, 32(6), 2010.

[70] Paul Green, Jan Kautz, and Frédo Durand. Efficient reflectance and visibility approximations for environment map rendering. In *Computer Graphics Forum*, volume 26, pages 495–502. Wiley Online Library, 2007.

[71] James Gregson, Michael Krimerman, Matthias B. Hullin, and Wolfgang Heidrich. Stochastic tomography and its applications in 3D imaging of mixing fluids. *ACM ToG*, 31(4):52:1–52:10, July 2012.

[72] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proc. CVPR*, 2013.

[73] Kai Han, Kwan-Yee K. Wong, and Miaomiao Liu. Dense reconstruction of transparent objects by altering incident light paths through refraction. *Int. J. Comput. Vision*, 126(5):460–475, May 2018.

[74] HDRI HAVEN. 100% free hdris, for everyone., 2017.

[75] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[77] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *CVPR*, 2019.

[78] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *CVPR*, 2017.

[79] Berthold K. P. Horn and Michael J. Brooks, editors. *Shape from Shading*. MIT Press, Cambridge, MA, USA, 1989.

[80] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*, 2016.

[81] Zhuo Hui and Aswin C. Sankaranarayanan. A dictionary-based approach for estimating shape and spatially-varying reflectance. In *ICCP*, 2015.

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

[83] C. P. Huynh, A. Robles-Kelly, and E. Hancock. Shape and refractive index recovery from single-view polarisation images. In *CVPR*, pages 1229–1236, June 2010.

[84] Ivo Ihrke, Kiriakos Kutulakos, Hendrik Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. *Comput. Graph. Forum*, 29:2400–2426, 12 2010.

[85] Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '04, pages 365–373, Goslar Germany, Germany, 2004. Eurographics Association.

[86] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[87] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J Mitra. Decomposing single images for layered photo retouching. In *Comput. Graph. Forum*, volume 36, 2017.

[88] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2017.

[89] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[90] Y. Ji, J. Ye, and J. Yu. Reconstructing gas flows using light-path approximation. In *CVPR*, pages 2507–2514, June 2013.

[91] M. K. Johnson and E. H. Adelson. Shape estimation in natural illumination. In *CVPR*, 2011.

[92] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters (RA-L)*, 2020.

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

[94] Yoshihiro Kanamori and Yuki Endo. Relighting humans: occlusion-aware inverse rendering for fullbody human images. *SIGGRAPH Asia*, 37(270):1–270, 2018.

[95] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 2013.

[96] Brian Karis and Epic Games. Real shading in unreal engine 4. *SIGGRAPH 2013 Courses: Physically Based Shading Theory Practice*, 2013.

[97] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[98] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):1, 2011.

[99] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3D object compositing. *ACM Transactions on Graphics*, pages 32:1–32:15, 2014.

[100] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[101] J. Kim, I. Reshetouski, and A. Ghosh. Acquiring axially-symmetric transparent objects using single-view transmission imaging. In *CVPR*, pages 1484–1492, July 2017.

[102] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 20–28, 2017.

[103] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[104] Martin Knecht, Georg Tanzmeister, Christoph Traxler, and Michael Wimmer. Interactive brdf estimation for mixed-reality applications. *WSCG*, 20(1), 2012.

[105] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[106] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3D shape by light-path triangulation. In *ICCV*, volume 2, pages 1448–1455 Vol. 2, Oct 2005.

[107] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218, 2000.

[108] Kiriakos N. Kutulakos and Eron Steger. A theory of refractive and specular 3D shape by light-path triangulation. *IJCV*, 76(1):13–29, 2008.

[109] Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *CVPR*, pages 5918–5928, 2019.

[110] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable Monte Carlo ray tracing through edge sampling. *ACM ToG (SIGGRAPH Asia)*, 37(6):222:1 – 222:11, 2018.

[111] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.

[112] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graphics*, 36(4), 2017.

[113] Z. Li, Z. Xu, R. Ramamoorthi, and M. Chandraker. Robust energy minimization for brdf-invariant shape from light fields. In *CVPR*, 2017.

[114] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *ECCV*, pages 371–387, 2018.

[115] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, 2020.

[116] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *ECCV*, 2018.

[117] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *ECCV*, pages 72–87, 2018.

[118] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia*, page 269. ACM, 2018.

[119] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, and Manmohan Chandraker. OpenRooms: An end-to-end open framework for photorealistic indoor scene datasets. In *CVPR*, 2021.

[120] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[121] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *ICCV*, 2017.

[122] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020.

[123] Stephen Lombardi and Ko Nishino. Reflectance and natural illumination from a single image. In *ECCV*, 2012.

[124] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM ToG (SIGGRAPH Asia)*, 38(4):65:1–65:14, 2019.

[125] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified shape and svbrdf recovery using differentiable monte carlo rendering. In *Computer Graphics Forum*, volume 40, pages 101–113. Wiley Online Library, 2021.

[126] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020.

[127] Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020.

[128] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[129] Julio Marco, Quercus Hernandez, Adolfo Munoz, Yue Dong, Adrian Jarabo, Min H Kim, Xin Tong, and Diego Gutierrez. Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Trans. Graphics*, 36(6), 2017.

[130] Stephen R Marschner, Stephen H Westin, Eric PF Lafortune, Kenneth E Torrance, and Donald P Greenberg. Image-based brdf measurement including human skin. In *Eurographics Workshop on Rendering Techniques*, pages 131–144. Springer, 1999.

[131] Stephen Robert Marschner. *Inverse rendering for computer graphics*. Cornell University, 1998.

[132] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. Graphics*, 22(3), 2003.

[133] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and rendering of transparent and refractive objects. In *Eurographics Workshop on Rendering*, EGRW '02, pages 267–278, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[134] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, 2017.

[135] Morgan McGuire, Michael Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2017.

[136] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *CVPR*, 2018.

[137] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[138] D. Miyazaki and K. Ikeuchi. Inverse polarization raytracing: estimating surface shapes of transparent objects. In *CVPR*, volume 2, pages 910–917 vol. 2, June 2005.

[139] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, June 2019.

[140] Nigel JW Morris and Kiriakos N Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[141] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic refraction stereo. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1518–1531, 2011.

[142] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, H-P Seidel, and Tobias Ritschel. Deep shading: convolutional neural networks for screen space shading. *Comput. Graph. Forum*, 36(4), 2017.

[143] Takuya Narihira, Michael Maire, and Stella X Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *ICCV*, 2015.

[144] Shree K. Nayar, Katsushi Ikeuchi, and Takeo Kanade. Shape from interreflections. *IJCV*, 6(3), 1991.

[145] Shree K. Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graphics*, 25(3), 2006.

[146] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. In *ACM transactions on graphics (TOG)*, volume 24, pages 536–543. ACM, 2005.

[147] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[148] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of brdf models. *Rendering Techniques*, 2005(16th):2, 2005.

[149] Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. On optimal, minimal brdf sampling for reflectance acquisition. *ACM Trans. Graphics*, 34(6), 2015.

[150] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D ken burns effect from a single image. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019.

[151] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering. 2021.

[152] Michael Oren and Shree K Nayar. Generalization of the lambertian model and implications for machine vision. *IJCV*, 14(3), 1995.

[153] Matthew O'Toole and Kiriakos N. Kutulakos. Optical computing for fast light transport analysis. *ACM Trans. Graphics*, 29(6), 2010.

[154] Geoffrey Oxholm and Ko Nishino. Shape and reflectance from natural illumination. In *ECCV*, 2012.

[155] Geoffrey Oxholm and Ko Nishino. Shape and reflectance estimation in the wild. *PAMI*, 38(2):376–389, 2016.

[156] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M Seitz. Photoshape: photorealistic materials for large-scale shape collections. *ACM Transactions on Graphics (TOG)*, 37(6):192, 2019.

[157] Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. Optix: a general purpose ray tracing engine. *Acm transactions on graphics (tog)*, 29(4):1–13, 2010.

[158] Pieter Peers and Philip Dutré. Wavelet environment matting. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 157–166, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[159] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[160] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[161] Julien Philip, Sébastien Morgenthaler, Michaël Gharbi, and George Drettakis. Free-viewpoint indoor neural relighting from multi-view stereo. *ACM Trans. Graph.*, 2021.

[162] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[163] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[164] Y. Qian, M. Gong, and Y. Yang. Frequency-based environment matting by compressive sensing. In *ICCV*, pages 3532–3540, Dec 2015.

[165] Y. Qian, M. Gong, and Y. Yang. Stereo-based 3D reconstruction of dynamic fluid surfaces by global optimization. In *CVPR*, pages 6650–6659, July 2017.

[166] Yiming Qian, Minglun Gong, and Yee-Hong Yang. 3D reconstruction of transparent objects with position-normal consistency. In *CVPR*, pages 4369–4377, 06 2016.

[167] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH*, 2001.

[168] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH*, 2001.

[169] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021.

[170] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Efstratios Gavves, and Tinne Tuytelaars. Deep reflectance maps. In *CVPR*, pages 4508–4516, 2016.

[171] Kosta Ristovski, Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, 2013.

[172] J. Riviere, P. Peers, and A. Ghosh. Mobile surface reflectometry. *Comput. Graph. Forum*, 35(1), 2016.

[173] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *arXiv preprint arXiv:2011.02523*, 2020.

[174] Fabiano Romeiro, Yuriy Vasilyev, and Todd Zickler. Passive reflectometry. In *ECCV*, 2008.

[175] Fabiano Romeiro and Todd Zickler. Blind reflectometry. In *ECCV*, 2010.

[176] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[177] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and SVBRDF estimation. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020.

[178] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv:1712.03931*, 2017.

[179] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[180] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–528, June 2006.

[181] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. *arXiv preprint arXiv:1901.02453*, 2019.

[182] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *CVPR*, 2018.

[183] Qi Shan, Sameer Agarwal, and Brian Curless. Refractive height fields from single and multiple images. In *CVPR*, pages 286–293, 06 2012.

[184] Evan Shelhamer, Jonathan T Barron, and Trevor Darrell. Scene intrinsics and depth from a single image. In *ICCV Workshop*, 2015.

[185] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martın-Martın, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, Kent Vainio, Li Fei-Fei, and Silvio Savarese. iGibson, a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint*, 2020.

[186] Jian Shi, Yue Dong, Hao Su, and X Yu Stella. Learning non-Lambertian object intrinsics across ShapeNet categories. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5844–5853. IEEE, 2017.

[187] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[188] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[189] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6918–6926, June 2019.

[190] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576. IEEE Computer Society, 2015.

[191] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017.

[192] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. *arXiv preprint arXiv:2012.03927*, 2020.

[193] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2020.

[194] Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. Single-shot analysis of refractive shape using convolutional neural networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 995–1003. IEEE, 2019.

[195] Jonathan Dyssel Stets, Alessandro Dal Corso, Jannik Boll Nielsen, Rasmus Ahrenkiel Lyngby, Sebastian Hoppe Nesgaard Jensen, Jakob Wilm, Mads Brix Doest, Carsten Gundlach, Eythor Runar Eiriksson, Knut Conradsen, Anders Bjorholm Dahl, Jakob Andreas Bærentzen, Jeppe Revall Frisvad, and Henrik Aanæs. Scene reassembly after multi-modal digitization and pipeline evaluation using photorealistic rendering. *Appl. Optics*, 56(27):7679–7690, 2017.

[196] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[197] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Trans. Graph.*, 38(4):79–1, 2019.

[198] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[199] K. Tanaka, Y. Mukaigawa, H. Kubo, Y. Matsushita, and Y. Yagi. Recovering transparent shape from time-of-flight distortion. In *CVPR*, pages 4387–4395, June 2016.

[200] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, 2018.

[201] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[202] Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic reconstruction of transparent objects. In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH, New York, NY, USA, 2006. ACM.

[203] C. Tsai, A. Veeraraghavan, and A. C. Sankaranarayanan. What does a single light-ray reveal about a transparent object? In *ICIP*, pages 606–610, Sep. 2015.

[204] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *TOG*, volume 25, pages 967–976. ACM, 2006.

[205] Eric Veach. *Robust Monte Carlo methods for light transport simulation*, volume 1610. Stanford University PhD thesis, 1997.

[206] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[207] Ting-Chun Wang, Manmohan Chandraker, Alexei Efros, and Ravi Ramamoorthi. SVBRDF-invariant shape and reflectance estimation from light-field cameras. *PAMI*, 2017.

[208] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12538–12547, 2021.

[209] Gregory J Ward. Measuring and modeling anisotropic reflection. *ACM Trans. Graphics*, 26(2), 1992.

[210] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, June 2016.

[211] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar. Refractive shape from light field distortion. In *ICCV*, pages 1180–1186, Nov 2011.

[212] Yonatan Wexler, Andrew Fitzgibbon, and Andrew Zisserman. Image-based environment matting. In *CVPR*, pages 279–290, 01 2002.

[213] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19, 1980.

[214] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Cong, and Hui Huang. Full 3D reconstruction of transparent objects. *ACM ToG*, 37(4):103:1–103:11, July 2018.

[215] Hongzhi Wu and Kun Zhou. Appfusion: Interactive appearance acquisition using a kinect sensor. *Comput. Graph. Forum*, 34(6), 2015.

[216] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3D environment. *arXiv preprint arXiv:1801.02209*, 2018.

[217] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[218] Zhaohui Wu, Zhong Zhou, Delei Tian, and Wei Wu. Reconstruction of three-dimensional flame with color temperature. *Vis. Comput.*, 31(5):613–625, May 2015.

[219] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[220] Fanbo Xiang, Zexiang Xu, Miloš Hašan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. *arXiv preprint arXiv:2103.00762*, 2021.

[221] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017.

[222] Hanggao Xin, Shaokun Zheng, Kun Xu, and Ling-Qi Yan. Lightweight bilateral convolutional neural networks for interactive single-bounce diffuse indirect illumination. *IEEE Annals of the History of Computing*, (1):1–1, 2020.

[223] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics (TOG)*, 32(6):209, 2013.

[224] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019.

[225] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal BRDF sampling for two-shot near-field reflectance acquisition. *ACM Trans. Graphics*, 35(6), 2016.

[226] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Trans. Graphics*, 37(4), 2018.

[227] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[228] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020.

[229] S. Yeung, T. Wu, C. Tang, T. F. Chan, and S. Osher. Adequate reconstruction of transparent objects on a shoestring budget. In *CVPR*, pages 2513–2520, June 2011.

[230] S. Yeung, T. Wu, C. Tang, T. F. Chan, and S. J. Osher. Normal estimation of a transparent object using a video. *PAMI*, 37(4):890–897, April 2015.

[231] Sai-Kit Yeung, Chi-Keung Tang, Michael S. Brown, and Sing Bing Kang. Matting and compositing of transparent and refractive objects. *ACM ToG (SIGGRAPH)*, 30(1):2:1–2:13, 2011.

[232] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020.

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

[234] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6), 2019.

[235] Edward Zhang, Michael F Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (TOG)*, 35(6):1–14, 2016.

[236] Hang Zhang, Kristin Dana, and Ko Nishino. Friction from reflectance: Deep reflectance codes for predicting physical surface properties from one-shot in-field reflectance. In *Proc. ECCV*, 2016.

[237] Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. Recovering scene geometry under wavy fluid via distortion and defocus analysis. In *ECCV*, volume 8693, pages 234–250, 09 2014.

[238] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *CVPR*, 2017.

[239] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. CVPR*, 2017.

[240] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. *ACM Transactions on Graphics (TOG)*, 30(4):1–10, 2011.

[241] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. *Proc. ECCV*, 2020.

[242] Hao Zhou, Xiang Yu, and David W. Jacobs. Glosh: Global-local spherical harmonics for intrinsic image decomposition. In *ICCV*, pages 7820–7829, October 2019.

[243] Hao Zhou, Xiang Yu, and David W Jacobs. Glosh: Global-local spherical harmonics for intrinsic image decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7820–7829, 2019.

[244] Jiayuan Zhu and Yee-Hong Yang. Frequency-based environment matting. In *Pacific Graphics*, pages 402–410, 2004.

[245] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 205–214, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[246] X. Zuo, C. Du, S. Wang, J. Zheng, and R. Yang. Interactive visual hull refinement for specular and transparent object surface reconstruction. In *ICCV*, pages 2237–2245, Dec 2015.