Lightweight material acquisition using deep learning

Valentin Deschaintre

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Acquisition légère de matériaux par apprentissage profond

Valentin Deschaintre

Inria Sophia Antipolis-Méditerannée, Optis an Ansys affiliate

Présentée en vue de l'obtention du grade de docteur en Informatique d'Université Côte d'Azur

Dirigée par : Adrien Bousseau, George Drettakis

Soutenue le : 25 Novembre 2019

Devant le jury composé de :

Pr. Holly Rushmeier, Professor, Yale University

Pr. Niloy Mitra, Professor, University College London

Pr. Matthias Nießner, Professor, Technical University of Munich

Pr. Xavier Granier, Professeur, Institut d'Optique

Dr. Abhijeet Ghosh, Associate Professor, Imperial College London

Dr. Adrien Bousseau, Chargé de Recherche, Université Côte d'Azur, Inria

Dr. George Drettakis, Directeur de Recherche, Université Côte d'Azur, Inria

Dr. Anthony Jouanin, Hardware & Measure manager, Ansys

Dr. Vincent Hourdin, Research engineer, Ansys

informatiques mathématiques Inria

ANSYS OPTIS

Acquisition légère de matériaux par apprentissage profond

Lightweight material acquisition using deep learning

Jury:

Président du jury / President of the jury
Pr. Holly Rushmeier, Professor, Yale University

Rapporteurs / Reviewers
Pr. Holly Rushmeier, Professor, Yale University
Pr. Niloy Mitra, Professor, University College London

Examinateurs / Examiners
Pr. Matthias Nießner, Professor, Technical University of Munich
Pr. Xavier Granier, Professeur, Institut d'Optique
Dr. Abhijeet Ghosh, Associate Professor, Imperial College London

Invités / Invited
Dr. Anthony Jouanin, Hardware & Measure manager, Ansys
Dr. Vincent Hourdin, Research engineer, Ansys

Directeurs de thèse / Thesis supervisors
Dr. Adrien Bousseau, Chargé de Recherche, Université Côte d'Azur, Inria
Dr. George Drettakis, Directeur de Recherche, Université Côte d'Azur, Inria

Acknowledgements

I would first like to express how grateful I am to my family for encouraging me to be curious and expanding my horizon throughout my youth. It is the support and education you provided me that made this whole adventure possible.

Of course, I would like to deeply thank Adrien Bousseau and George Drettakis, my advisors, for their support, encouragements and guidance during my PhD. I am also grateful to you for the freedom to explore the directions I was curious about, for introducing me into an incredible community and for the opportunities you made possible that I would never have hoped for.

One of these opportunities was to collaborate with MIT, and I thank Fredo Durand and his whole group very much for welcoming me during my visits there!

My thesis was funded through the French CIFRE system and I am grateful to Optis, Inria and ANRT for making this possible. Especially Vincent Hourdin for being there for me from the first moment I drafted the idea of this thesis and Anthony Jouanin for supervising the industrial aspect of my work.

But these years would have been lonely if not for the Graphdeco group members, I thank you all for our debates and laughs during lunches, coffee breaks and events, the technical discussions and philosophical research conversations.

In particular, I would like to mention Simon Rodriguez with whom I shared not only the office but also the ups and downs inherent to research, and many jokes! I also thank Yulia Gryaditskaya very much for her support, her kindness and our discussions which helped me to better understand the academic world and encouraged me to explore it further. I am also grateful to Miika Aittala for his help in building my knowledge, for teaching me some Finnish, but most importantly for our never ending chats around MIT's couches.

I met too many great people during these years to list exhaustively, but if we exchanged ideas, had dinner, drank a beer or simply shared tea, these acknowledgements are also meant for you!

Résumé

Que ce soit pour le divertissement ou le design industriel, l'infographie est de plus en plus présente dans notre vie quotidienne. Cependant, reproduire une scène réelle dans un environnement virtuel reste une tâche complexe, nécessitant de nombreuses heures de travail. L'acquisition de géométries et de matériaux à partir d'exemples réels est une solution, mais c'est souvent au prix de processus d'acquisitions et de calibrations complexes. Dans cette thèse, nous nous concentrons sur la capture légère de matériaux afin de simplifier et d'accélérer le processus d'acquisition et de résoudre les défis industriels tels que la calibration des résultats. Les textures et les ombres sont quelques-uns des nombreux indices visuels qui permettent aux humains de comprendre l'apparence d'un matériau à partir d'une seule image. La conception d'algorithmes capables de tirer parti de ces indices pour récupérer des fonctions de distribution de réflectance bidirectionnelles (SVBRDF) variant dans l'espace à partir de quelques images pose un défi aux chercheurs en infographie depuis des décennies. Nous explorons l'utilisation de l'apprentissage profond pour la capture légère de matériaux et analyser ces indices visuels. Une fois entraînés, nos réseaux sont capables d'évaluer, par pixel, les normales, les albedos diffus et spéculaires et une rugosité à partir d'une seule image d'une surface plane éclairée par l'environnement ou un flash tenu à la main. Nous montrons également comment notre méthode améliore ses prédictions avec le nombre d'images en entrée et permet des reconstructions de haute qualité en utilisant jusqu'à 10 images d'entrées — un bon compromis entre les approches existantes.

Mots-clés: Acquisition de matériaux - Apprentissage profond

Abstract

Whether it is used for entertainment or industrial design, computer graphics is ever more present in our everyday life. Yet, reproducing a real scene appearance in a virtual environment remains a challenging task, requiring long hours from trained artists. A good solution is the acquisition of geometries and materials directly from real world examples, but this often comes at the cost of complex hardware and calibration processes. In this thesis, we focus on lightweight material appearance capture to simplify and accelerate the acquisition process and solve industrial challenges such as result image resolution or calibration. Texture, highlights, and shading are some of many visual cues that allow humans to perceive material appearance in pictures. Designing algorithms able to leverage these cues to recover spatially-varying bi-directional reflectance distribution functions (SVBRDFs) from a few images has challenged computer graphics researchers for decades. We explore the use of deep learning to tackle lightweight appearance capture and make sense of these visual cues. Once trained, our networks are capable of recovering per-pixel normals, diffuse albedo, specular albedo and specular roughness from as little as one picture of a flat surface lit by the environment or a hand-held flash. We show how our method improves its prediction with the number of input pictures to reach high quality reconstructions with up to 10 images — a sweet spot between existing single-image and complex multi-image approaches — and allows to capture large scale, HD materials. We achieve this goal by introducing several innovations on training data acquisition and network design, bringing clear improvement over the state of the art for lightweight material capture.

Keywords: Material acquisition - Deep learning

Contents

Chapter 1

Introduction

In this section I provide a brief introduction to the topic of this thesis, I then summarize the main challenges and our key contributions. I conclude this chapter with the overview of how the research conducted over my PhD is supplemented with a contribution to industrial challenges.

1.1 Motivation

Photo-realism has been pursued in Computer Graphics for decades. With better algorithms and more computing power, it is now possible to render scenes that are virtually impossible to distinguish from a photograph (Figure 1.1).

Nevertheless, creating realistic virtual scenes or characters takes hours for trained artists. We ask the following questions in this thesis, and attempt to provide first answers: can we facilitate this process? Can we provide the tools which would allow non-expert users to create their own virtual content?

In this thesis we make an important step towards achieving this goal. We build on the idea that real scenes are a rich source of information that can be exploited for virtual scene creation. This will not only reduce the workload in the movie and games industries, but also facilitate every-day use of virtual content. For instance, one might be interested in recreating their living space in order to experiment with lighting, wallpapers, furniture, other design elements, or integrate familiar elements in larger scenes.

Real scene reproduction in a virtual environment is one of the big challenges at the intersection between computer vision and computer graphics. To ensure an immersive and convincing experience, the appearance of a scene has to be re-created accurately. This is a challenging task, as the appearance of real-world objects results from complex interactions between light, material reflectance, and geometry. While scene geometry can be obtained with techniques such as multiview stereo algorithms or depth scanners, recovering a numerical representation of materials from photographs or measurements

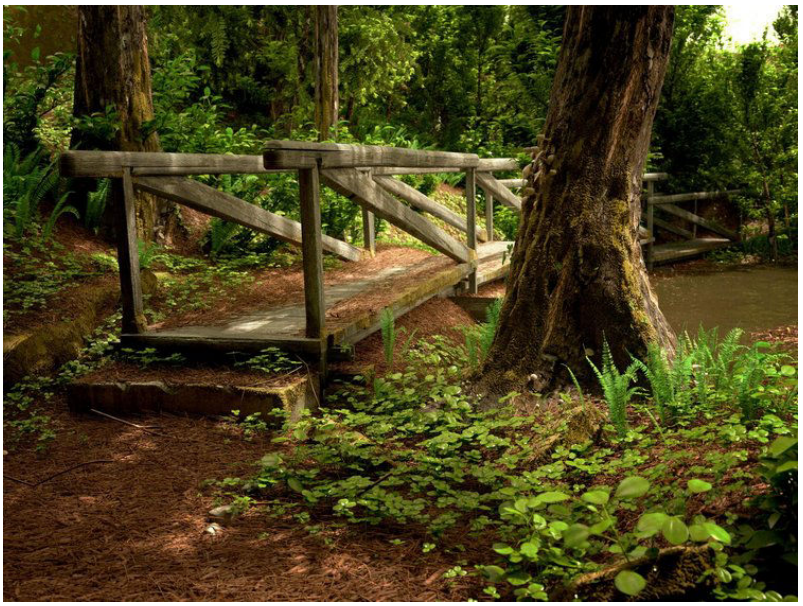


Figure 1.1: A photo-realistic rendering which won the 2012 blenderGuru Photorealism Competition. Author: Major4z

of a surface is at the heart of *appearance capture* algorithms, that is the focus of this thesis.

The visual appearance and an accurate physical representation of materials are important in multiple industries. In industrial design, for example, it reduces the need for physical prototypes by allowing to quickly and efficiently test different appearances and to observe materials behavior under different lighting conditions. For instance, accurate reproduction of materials is crucial for the car industry, where reflectance properties of the materials have to be carefully taken into consideration before production (as illustrated in Figure 1.2).

Virtual content is also widely used in movie making, since it not only allows to create entirely virtual environment, but also to reduce the shooting costs in difficult environments, or conditions in terms of teams and equipment. Furthermore, it allows to create multiple versions of the content and facilitates its editing. Figure 1.3 illustrates how such techniques are used in recent movies. Visually consistent integration of characters, objects and visual effects is key to engaging users and maintaining their attention. This cannot be achieved without convincing material appearance.

Figure 1.2: An example of virtual prototyping with Ansys software. Image source: www.3dprintingmedia.network

Figure 1.3: The top part shows the scene capture by cameras, in contrast with the bottom part, showing the addition of Godzilla and the bridge. Image credit: Godzilla, ©Warner Bros, 2014

1.2 Challenges and contributions

Traditionally, accurate material appearance capture requires dense sampling of light and view directions in a controlled acquisition enviroment [SSW+14, XNY+16]. Such an approach has been used in production for movie making for example. Expensive light stages based on the design by Debevec et al. [DHT+00] were used for movies like Avatar (2009) or Superman Returns (2006).

Such advanced capture setups are needed because many different reflectances, geometries or lighting can yield the same observed image. For example, any single photograph can be reproduced by a diffuse albedo map, where highlights are "painted" over the surface. Nevertheless, even given a single image, human observers can immediately understand the material in many cases thanks to our prior knowledge.

Designing *a priori* assumptions about the space of plausible material solutions to guide simpler acquisition processes has challenged researchers for decades [GGG+16]. Recently, Deep Learning has emerged as a powerful method to automatically *learn* effective priors from data.

In this thesis, we focus on the challenges of *lightweight material capture* and propose supervised deep learning approaches. We train neural networks to solve the ill-posed *inverse* problem of estimating a spatially-varying bi-directional reflectance distribution function (SVBRDF) from a limited number of pictures. SVBRDFs represent the behavior of materials, depending on the incoming light and the view direction. We use the Cook-Torrance [CT82] analytical material model, allowing to represent each pixel of a SVBRDF with just a few parameters. We chose this model for its capacity to represent many classes of materials and its wide use in industry.

To achieve high performance with Deep Learning approaches, one has to build a tailored network architecture and loss function. Moreover, supervised learning requires a large amount of representative training data. Acquiring such data is a common challenge of deep-learning based approaches. I summarize below our solutions to these challenges and detail them in the following chapters.

Architecture design

The task of our deep networks is to predict four maps corresponding to *per-pixel* Cook-Torrance [CT82] parameters representing the material appearance. Importantly, this model distinguishes the *Diffuse* albedo, representing the light that is reflected in all directions by the material, and the *Specular* albedo, representing the light reflected around the mirror direction.

In this thesis I present a number of novel algorithms allowing the capture of a plausible material representation using different lightweight acquisition setups. Our algorithms adapt to the available input, whether it is a single flash picture, multiple ones, or a large scale image.

Single-image acquisition. In Chapter 3 we propose an algorithm that requires one single near-field flash-lit photograph as input. Flash photographs are easy to acquire, and have been shown to contain a lot of information that can be leveraged in inferring the material properties from one [AP07, AWL15, AAL16] or multiple images [RPG16, HSL+17]. In such images, the pixels showing the highlight provide strong cues about specularity, whereas the outer pixels show diffuse and normal variations more prominently. To arrive at a consistent solution across the image, these regions need to share information about their respective observations.

Our experiments reveal that existing deep learning architectures struggle to aggregate distant information in an image. To address this limitation, we develop a secondary network that extracts global features and propagates it through the main network, facilitating back-and-forth exchange of information across distant image regions.

Multiple-image acquisition. Our methods allow to retrieve a SVBRDF representation from just a single flash picture, but in many cases a single photograph simply does not contain enough information to completely acquire a material. In Chapter 4, we propose a method that is capable of aggregating the cues provided by additional pictures, while retaining a lightweight capture procedure. With this method, we are able to improve the results with more pictures until the user is satisfied with the result. We discuss both the quantitative and qualitative improvement of the results as they get closer to ground truth with more inputs photographs.

Acquisition scale, user control and high resolution. The solutions proposed in Chapters 3 and 4 —and concurrent work [LSC18, GLD+19] are nonetheless limited in terms of resolution, scale of acquisition and user control. Indeed, near-field flash lighting greatly restricts the *scale* at which materials can be captured – typically around twenty centimeters wide using a cell phone held at a similar distance. Another common limitation of the above methods is that they rely on black-box optimization or deep learning to infer SVBRDF parameters from few measurements, offering little *user control* on their output. We address these two challenges in Chapter 5 by proposing a *by-example*, *multi-scale* appearance capture method, which recovers SVBRDF parameter maps over large-scale environmentally lit surfaces by propagating information from a few small-scale *exemplar SVBRDF patches*.

An added benefit of using environmental lighting is that we can split the large-scale image into smaller tiles processed independently by our deep network, using the same exemplars to promote coherence. This mechanism allows us to treat high-resolution images as collections of tiles that we stitch seamlessly in a post-process, resulting in SVBRDF maps of up to 4K pixels wide. In contrast, existing deep learning methods need to process the input images in their entirety to exploit the complementary visual cues given by the spatially-varying flash lighting, which limits these methods to small resolutions to fit in GPU memory [DAD+18, LSC18, DAD+19, GLD+19].

Through our work, we show that our deep learning based SVBRDF acquisition methods are able to produce convincing results for complex spatially varying materials made from multiple elements.

Data generation

To predict material models, we use supervised learning, which comes with the challenge of finding enough training data. We solve the lack of large real-world material datasets by leveraging artist-created, procedural SVBRDFs [All19b], which we sample and render under multiple lighting directions to create training images. We further augment the data by randomly mixing these SVBRDFs together (Chapter 3) and introducing an in-line rendering pipeline (Chapter 4), allowing for virtually infinite material variation.

Loss design

To train a network a suitable loss function must be defined, which evaluates the quality of the output model parameters against the ground truth. A naive optimization function (loss) would be to directly minimize the pixel-wise difference between predicted and ground truth material model. But this approach is suboptimal, as it does not consider the interactions between the different material parameter maps. Intuitively, while a predicted map may look plausible when observed in isolation, it may yield an image far from the ground truth when combined with the other parameter maps when evaluating the BRDF function.

Furthermore, the numerical differences in the parameter maps might not consistently correlate with differences in the material's appearance, causing the naive loss to weight the importance of different features arbitrarily. We mitigate these shortcomings by formulating a differentiable SVBRDF similarity metric that compares the *renderings* of the predicted maps against renderings of the ground truth from several lighting and viewing directions.

1.3 Thesis Context

This thesis was funded by a CIFRE (Academic/Industrial) collaboration between the French government (ANRT), Inria and Optis, an ANSYS affiliate. We therefore took industrial challenges into account during the definition of our research axes. We describe the experiments we conducted to address some of these challenges, while respecting the required industrial confidentiality.

Optis is a software development company specialized in physics based lighting simulation. More specifically, Optis proposes solutions for optical simulation, physics-based rendering, virtual reality and physics-based sensor simulation for industry. This allows engineers and artists to prototype in a virtual environment and quickly iterate to not only improve visual design, but also the assembly process, potential visual discomfort or usability. This process limits the need for costly physical prototypes and reduces development time.

To physically simulate light behavior, Optis identified a need in accurately measuring materials. The Hardware & Measure team is responsible for the evolution of the two Gonioreflectometers developed at Optis. The OMS2 is portable and capable of measuring

a BRDF in around a minute, while the OMS4 measure bench focuses on higher precision and volumetric measurements. Both devices only provide one BRDF per material and do not measure their spatial variations.

With lightweight acquisition, this thesis complements Optis' expertise in material acquisition. Our results simplify the process for artists to evaluate multiple materials during the design phase or to recreate a familiar environment. With this, the heavy and long measurement with specialized tools can be postponed to a phase where the product is better defined.

In addition, my responsibilities during this thesis included knowledge transfer and adaptation of the academic research to industrial needs. In Chapter 6, I describe some of the work done to answer to some of the issues encountered, such as inferring a different material model or evaluating our method against real measurements.

Finally, in the conclusion of this thesis, I discuss our methods, results and provide interesting directions to explore for future work.

Chapter 2

Related Work

2.1 Materials and rendering

2.1.1 Rendering

The goal of a rendering pipeline is to simulate the appearance of a 3D scene, given information about camera positions, scene geometry, surface materials and lighting conditions. For the purpose of this thesis, I focus on realistic rendering systems aiming to reproduce real world appearance. A rendering is obtained by simulating light interaction with objects in a 3D environment, as defined by the rendering equation [Kaj86]:

$$L_o(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + \int_\omega f_r(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \lambda, t)(\omega_i \cdot n) d\omega_i$$

L_o represents the radiance depending on the wavelength λ, the location in space x, the outgoing light direction ω_o and time t. L_e defines the emitted radiance, while the integral over ω represents the interaction between the surface material and the incoming light contributions over the hemisphere centered around the normal n to the surface. More specifically, f_r represents the material reflective properties, function of the incoming light direction ω_i, ω_o, x, t and λ . L_i represents the incoming light as a function of position x, ω_i, t and λ. Finally $(\omega_i \cdot n)$ represents the attenuation factor of influence of a light source due to incident angle. Intuitively, this equation represents how the occlusion of light, orientation and distance of different objects drives the shading and how the different materials in the scene drive light reflection, refraction or transmission. Multiple rendering methods simulate the light behavior with various degrees of accuracy and computational power required to produce a frame. This thesis focuses on the acquisition of the material reflective properties f_r from few images.

2.1.2 Materials

During the rendering process, light interacts with different surfaces, each made of different materials. For each interaction, the renderer needs to resolve which part of the

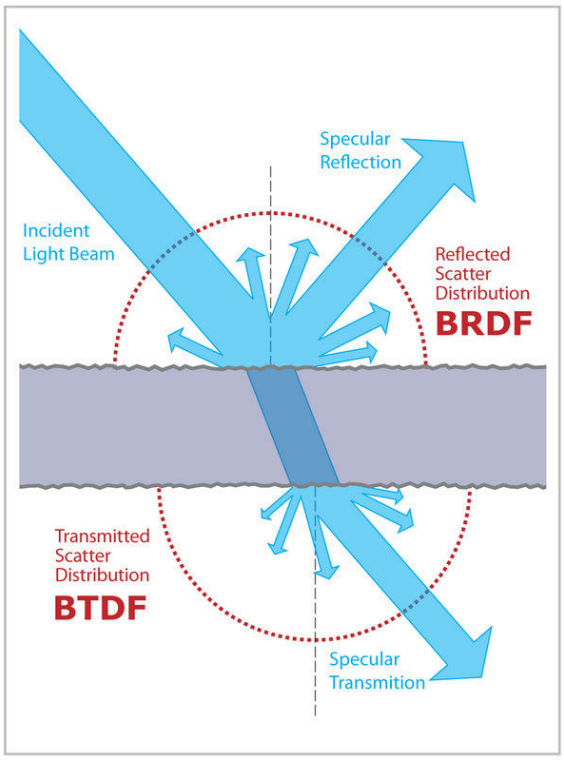


Figure 2.1: Light effects represented by a BRDF and a BTDF. The BRDF represents the reflected part of light, while the BTDF represents the transmitted part. Credit: Wikipedia User:Jurohi

light is absorbed, transmitted and reflected. Bartel et al. [FEW81] define the Bidirectional scattering distribution function (BSDF) to represent both the transmittance (BTDF) and the reflectance (BRDF) of a material —f_r in Kajiya rendering equation— , see Figure 2.1.

2.1.3 Material representation

In this thesis we focus on the reflection and absorption properties of materials. The most common representation is the Bidirectional Reflectance Distribution Function (BRDF) which defines for each incident lighting angle (θ_i, ϕ_i) and outgoing view angle (θ_o, ϕ_o) the amount of energy that is absorbed or reflected for each considered wavelength λ. The BRDF is an approximation of the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF) described by Nicodemus et al. [NRH+77]. With a BRDF, it is assumed that all light enters and leave from the same point, while BSSRDFs allow to model light behavior below material surfaces -such as in human skin. The Spatially

Figure 2.2: Examples of SVBRDFs rendered using a parametric model and a virtual environment.

Varying BRDF (SVBRDF) adds two spatial dimensions, allowing to define maps of varying appearance (e.g. for multi-material objects). Figure 2.2 shows renderings generated using SVBRDFs and a virtual environment.

A different approach to material appearance representation was introduced by Dana et al. [DVGNK99] in the form of Bidirectional Texture Function (BTF). This representation consists of hundreds to thousands of different pictures of a texture from different light/view angle. The rendering process samples for each different pixel the closest view/light configurations available in the pictures, matching the rendering condition. Unlike the BRDF, this representation includes meso-structures —small geometric details on the surface—, shading, sub-surface scattering, cast shadows and other subtle effects visible on the material as they are present in the acquired pictures. This capture of a wide variety of effects comes with the cost of a complex acquisition setup -see Figure 2.3), a challenging interpolation between pictures for novel views and a significant amount of data for each material. This representation is therefore not suited for our lightweight acquisition problematic. Nevertheless, these challenges are addressed

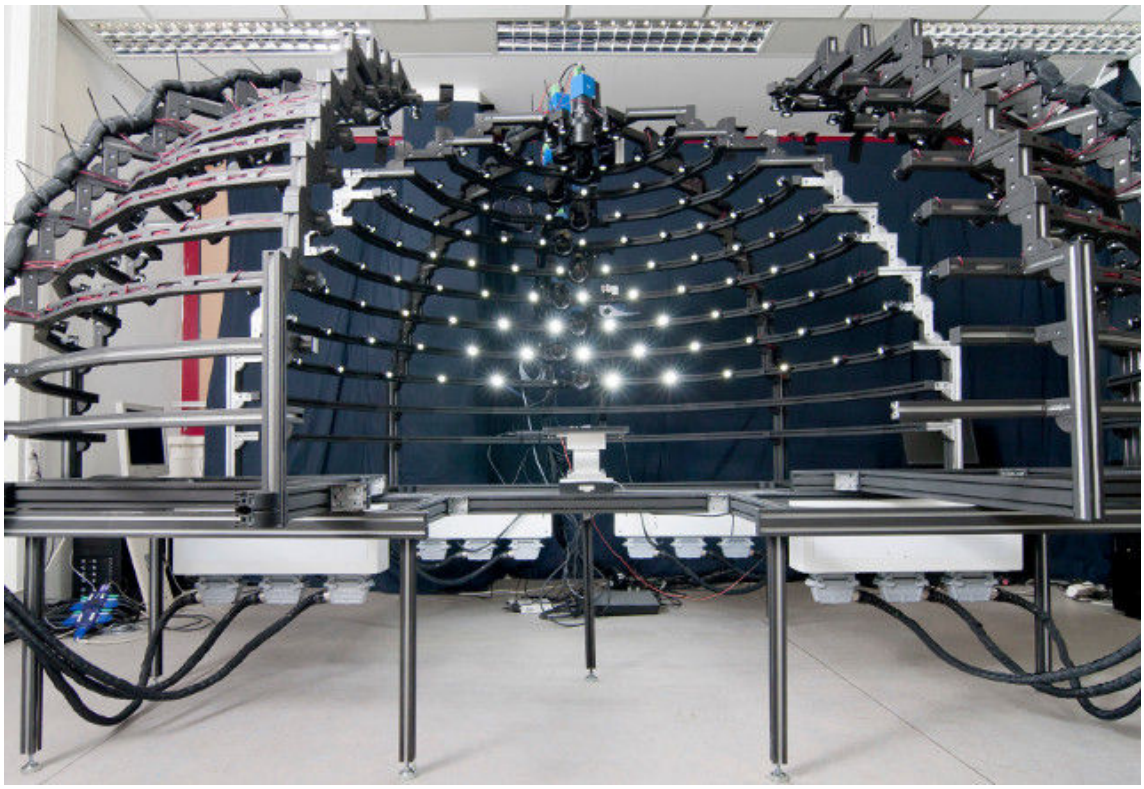


Figure 2.3: The Dome II, a BTF acquisition device developped by the University of Bonn [SSW+14].

in work such as Rainer et al. [RJGW19], Wu et al. [WDR11], Havran et al. [HFM10] or Ruiters et al. [RK09].

While we use BTFs for comparison and evaluation of our work, our main material representation in this thesis are SVBRDF.

2.1.4 Material models

Measured BRDFs are usually acquired in a tabulated representation. The percentage of reflected light is stored for each θ_i, ϕ_i, θ_o, ϕ_o and wavelength λ. This results in precise depiction of material interaction with light, but leads to large amounts of data and therefore limits the edition possibilities —as five dimensional arrays can prove difficult to navigate. For more compact representations, material models approximate real world

material behavior using mathematical functions. These models are separated in two main categories, empirical -also called phenomenological- and physically-based models.

The empirical models represent the appearance of a material with arbitrary parameters; they do not rely on the underlying physics of light behavior. Empirical material models include Phong [Pho75] or Ward [War92] for example. The Phong model uses 3 parameters: diffuse, specular power and specular exponent. It is a simple model allowing to efficiently compute the appearance of simple diffuse or specular materials. As an empirical model it does not aim at physical realism, for example, it does not respect energy conservation and reciprocity. In 1992, the Ward [War92] model was proposed to be an intermediate between theoretical models, too complex to efficiently render, and fully empirical model such as Phong [Pho75]. The proposed model is "physically plausible", satisfying energy conservation at most angles and reciprocity, while still being based on empirical data.

On the opposite end of the spectrum, physically based models attempt to derive a meaningful set of parameters from physical theory. Examples of such models are Beckmann [BS87], Torrance-Sparrow [TS67] or Cook-Torrance [CT82]. The Beckmann model is based on electromagnetic laws and rough surface modelling. On the other hand the Torrance-Sparrow and Cook-Torrance are base on geometric optics and use a micro-facets surface model. Each surface is composed of microscopic surfaces, oriented with respect to the general surface normal. The orientations are defined by a probability distribution function, varying between the different methods. In our work, we use the GGX distribution[WMLT07] of the Cook-Torrance model.

The GGX model is driven by three parameters. The first and second are diffuse and specular albedos which control the color and intensity of the light of the diffuse and specular behaviors at the interface with the material. The third is the roughness, which defines how glossy a material is by acting on the micro-facets orientation distribution. Examples of materials parametrized with this model and their renderings is available in Figure 2.4. The exact equations we use in this thesis are described in the form of pseudo-code in Appendix A.1. Figure 2.4 shows examples of parameter maps and their associated renderings.

More details on the many different material representations and models are discussed in the extensive survey by Guarnera et al. [GGG+16].

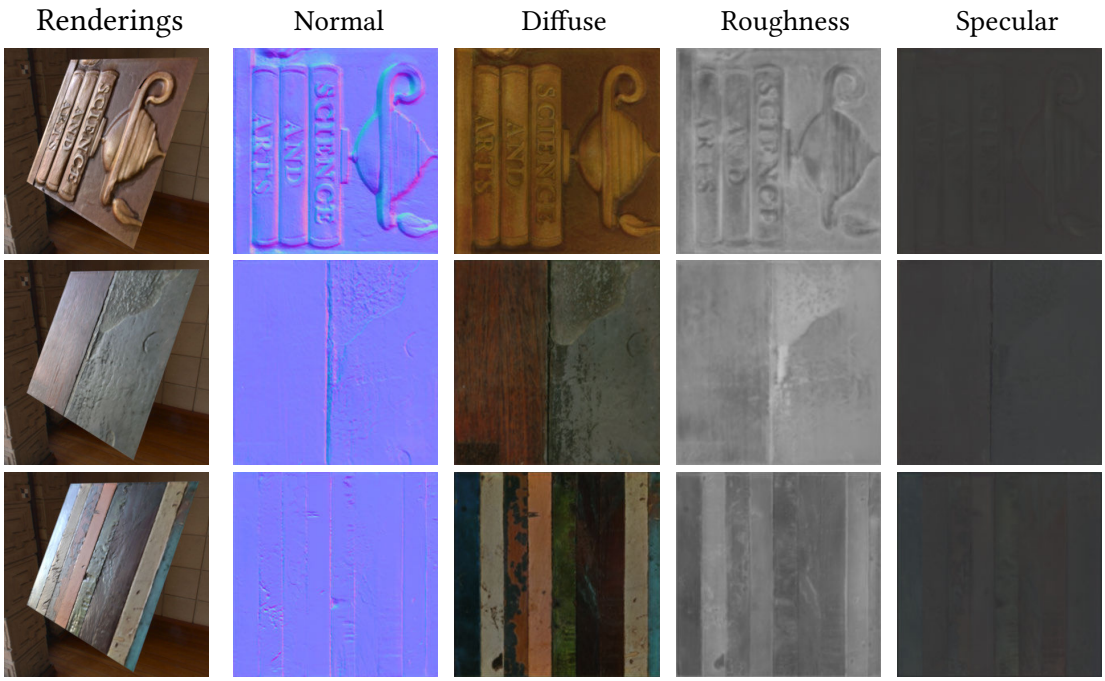


Figure 2.4: Materials represented by their parametrization in the GGX [WMLT07] model and the associated renderings. More details on the meaning of parameters are available in section 2.1.4.

2.1.5 Material synthesis

Given the complex interaction between parameters during rendering, the generation of a material with the desired appearance is a complex process. One option is to use specialized software such as Substance Designer [All19a] or Quixel [Qui19] for artistic design of a material. This solution allows for flexible creation and edition, but requires hours of work from a highly trained artist to achieve the desired appearance. Figure 2.5 shows a typical graph-based nodal representation for the creation of a procedural material.

In this context, the reproduction of a real world material appearance is complex, as it requires a careful evaluation of its appearance and physical properties.

2.1.6 Material Acquisition

A solution to reproduce a real-world material in a more systematic way is acquisition. Many different methods have been proposed with various degrees of complexity and precision.

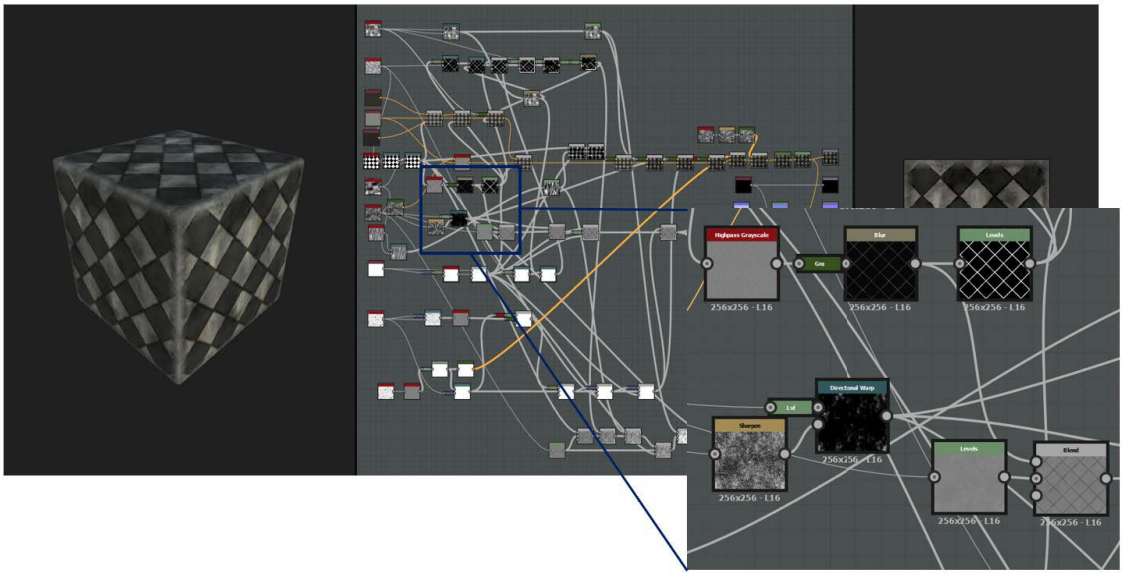


Figure 2.5: Screenshot from the nodal interface of Substance Designer [All19a]. Designing a material requires the creation of a complex graph of nodes.

2.1.6.1 Complex material acquisition

Early acquisition systems are aimed at exhaustively measuring a material in all possible configurations of light and view directions. Gonioreflectometer designs were proposed by Nicodemus et al. [NRH+77], Hsia and Richmond [HR76], Murray-Coleman and Smith [MCS90] or Ward [War92] among others. Further efforts focused on capturing appearance under controlled view and lighting conditions, first using motorized point lights and cameras [Mca02, DVGNK99] and later using complex light patterns such as linear light sources [GTHD03], spherical gradients [GCP+09], Fourier basis [AWL13], or deep-learned patterns [KCW+18].

Specialized hardware are developed for joint recovery of the SVBRDF and geometry [HLZ10] and simpler compact BRDF acquisition device combined with higher scale acquisition of diffuse appearance [DWT+10].

More conveniently, the work of Riviere et al. [RPG16] uses semi-controlled, hand held consumer grade hardware. While simplifying the acquisition process, the method still requires a couple of hundred pictures to achieve good results.

These methods provide high-quality capture of complex material effects —including anisotropy—, but they require tens to hundreds of measurements acquired, often using dedicated hardware.

2.1.6.2 Lightweight material acquisition

While complex acquisition hardware is reserved to a small, professional elite, lightweight material acquisition aims at providing simpler methods, using consumer level hardware, often at the cost of some precision. A good example is the work by Ren et al [RWS+11], simplifying the use of a linear light source as proposed by Gardner et al. [GTHD03]. This method is based on a simple smartphone and hand held linear light source in combination with a set of known BRDF arranged in a chart visible in the pictures (Figure 2.6). This illustrates the aforementioned trade-off of convenience against precision, simplifying the acquisition setup and process while making the assumption that the acquired SVBRDF is a combination of these provided set of references. Indeed, with less input information the problem becomes ill-posed and requires priors about the acquired material, the acquisition setup or environment.

In the case where only a few measurements of the material are available, a number of assumptions have been proposed to reduce ambiguity. Common priors include spatial and angular homogeneity [ZREB06] to exchange spatial resolution of pictures for higher angular resolution, repetitive or random texture-like behavior [WSM11, AWL15, AAL16] to leverage the lighting gradient, sparse environment lighting [LN16, DCP+14] allowing for better lighting condition reconstruction, polarization of sky lighting [RRFG17] to separate diffuse and specular behaviors, mixture of basis BRDFs [RWS+11, HSL+17] for material matching, optimal sampling directions [XNY+16] to maximize the information available in the measurements, and user-provided constraints such as rough global shading or reflectance information [DTPG11]. However, many of these assumptions restrict the family of materials that can be captured. For example, while the method by Aittala et al. [AAL16] takes a single flash image as input, it cannot deal with non-repetitive material samples.

A more detailed description of the materials acquisition methods up to 2015 is available in the Guarnera et al.[GGG+16] survey.

In contrast to these methods, we do not want to manually design priors, potentially

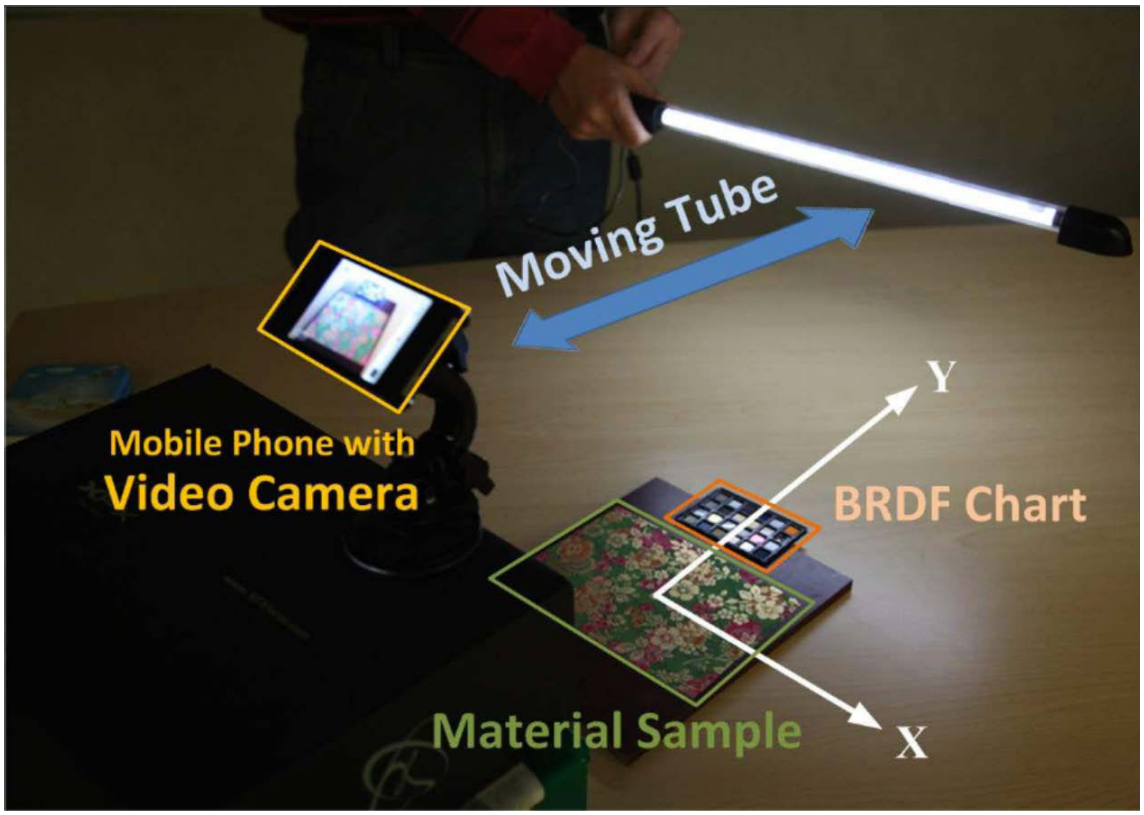


Figure 2.6: Capture setup of the method presented by Ren et al. [RWS+11] using a linear source and a set of known BRDF.

limiting the scope and applicability of our method. We use deep learning to train a network, automatically building priors directly from data.

2.2 Deep learning

Deep learning is a Machine learning technique, based on neural networks, designed to learn how to solve complex tasks from large amounts of data. Computational Neural networks were introduced by McCulloch & Pitts [MP43] proposing a model based on the idea that neural events can be treated by means of propositional logic. By combining multiple simple "logical" units, it is possible to represent complex expressions. Later on, the perceptron was introduced by Rosenblatt [Ros58] as a binary classifier neuron. Given inputs $(x_1 ... x_n)$, weights $w_1 ... w_n$ and a bias b, the perceptron output is defined by

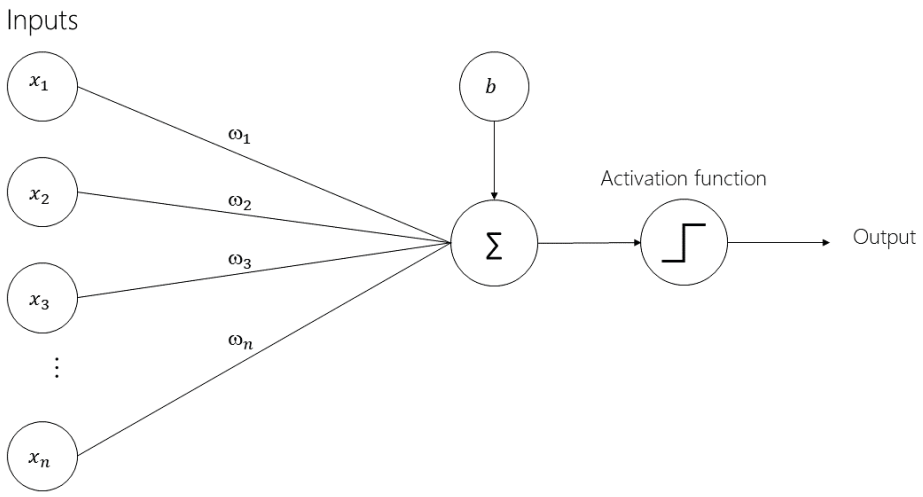


Figure 2.7: Single layer perceptron network. All inputs are directly connected to the output.

its activation function and will traditionally return 1 if

$$\sum_{i=1}^{n} w_i x_i + b >= 0$$

, 0 otherwise. Originally organized in a single layer, see Figure 2.7, with inputs directly connected to the output, a multi-layer version -Figure 2.8, using the chain rule to define a back propagation protocol was introduced by Rumelhart et al. [RHW86].

The processing of the input through the network is called a forward pass. The inverse, going from the output to the inputs of the network is defined as the backward pass. The back propagation of error uses the chain rule to define the gradient for each neuron during this backward pass with respect to the previous forward pass output. An optimization process such as gradient descent then uses these gradients to adjust the weights and bias associated to each neuron based on a differentiable optimization function —also called ”loss”. This loss function is central to the success of the optimization process. In this thesis, I show how problem specific knowledge is essential to designing

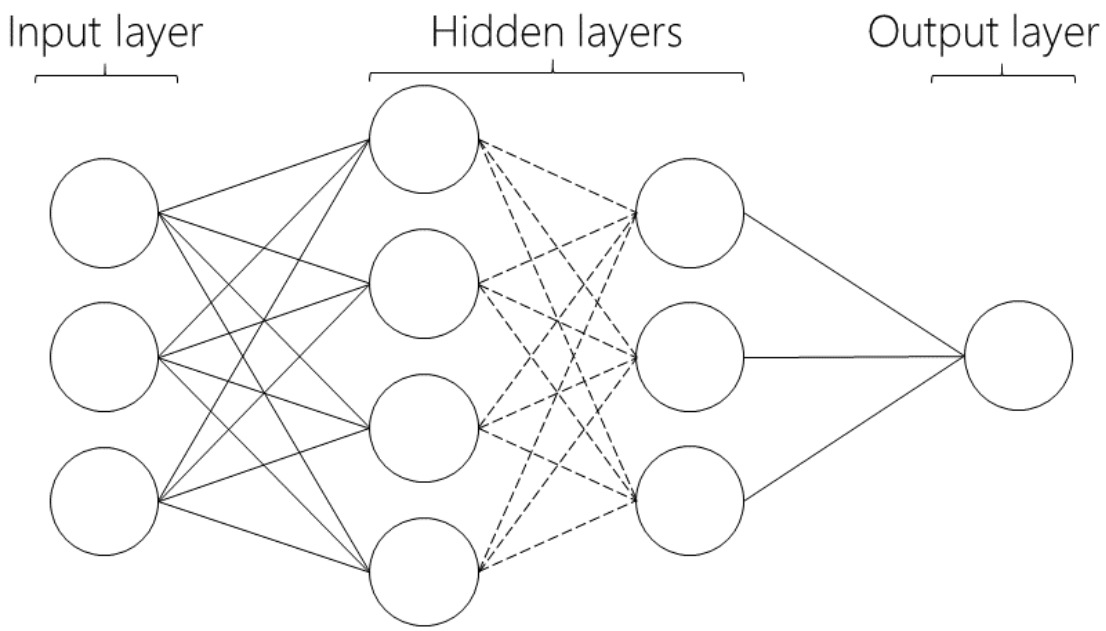


Figure 2.8: Multi layer network. Hidden layers are inserted between the inputs and the outputs, allowing for more complexity in the inference process. With more neurons comes the need to fine tune each of their weights and bias, this is where the back-propagation of error is crucial.

a loss capable of guiding the optimization process to the best solution space, leading to significantly improved results compared to generic L1 function for example.

Based on the multi-layer networks, Lecun et al. [LBD+90] introduced the convolutional neural network. With this new architecture, the neurons, which were previously all inter connected from a layer to another, are only connected to a sub-part of the previous layer in a sliding window fashion. With this improvement, it became possible to drastically reduce the number of parameters and connections between neurons, allowing the technology to scale better to higher resolution input data. The use of a sliding window approach also made the features extracted by the deep network spatially invariant as each "window" of the input to a layer of neurons will be treated similarly.

Recent deep networks are therefore composed of a combination of convolutional and fully connected layers with millions of interconnected neurons. The training process is

based on large datasets providing sufficient information for the optimization process to tune the weights and bias of the network through many iterations of back propagation. At inference time, we use the trained weights and architecture to quickly process new inputs.

2.2.1 Classical architectures

In recent years, deep learning has proven to be an efficient solution for a variety of problems such as image processing [KSH12], translation [SVL14], speech recognition [HDY+12] or geometry processing [QSMG16]. In 2012 Krizhevsky et al [KSH12] published AlexNet and won the ImageNet [DDS+09] contest by an important margin. Since then, many architectures were proposed opening new applications for deep learning, I will now briefly describe a few that strongly influenced the domain.

AlexNet is a classification network made of an encoder. The input picture is passed through multiple layers, gradually reducing the resolution, creating an internal representation -also called latent vector. In the case of classification, the last network layers transforms the internal representation in a probability distribution over all the possible classifications.

He et al. [HZRS16] introduced ResNet to expand the depth of deep networks, tackling the vanishing/exploding gradient [BSF94, GB10] problem by facilitating information flow between co-located features on different layers through "skip connection", which makes the training better behaved.

Adding to the encoder used to process and encode multiple scale information, the U-Net [RPB15] architecture uses a decoder to expand the results back to the original image resolution, allowing to solve image to image transformation problems such as image segmentation. Skip connections are used to propagate high frequency details through the architecture by directly connecting layers with the same resolution in the encoder and decoder. Figure 2.9 describes the architecture in more details.

Generative Adversarial Nets(GANs) were proposed by Goodfellow et al.[GPAM+14], allowing to generate new data through an adversarial network, judging the likelihood that the generator network output is real or generated. A combination of U-Net and GAN architecture was proposed by Isola et al. [IZZE17]. While this architecture shows good result on many image to image translation problems, we didn't find the GAN component

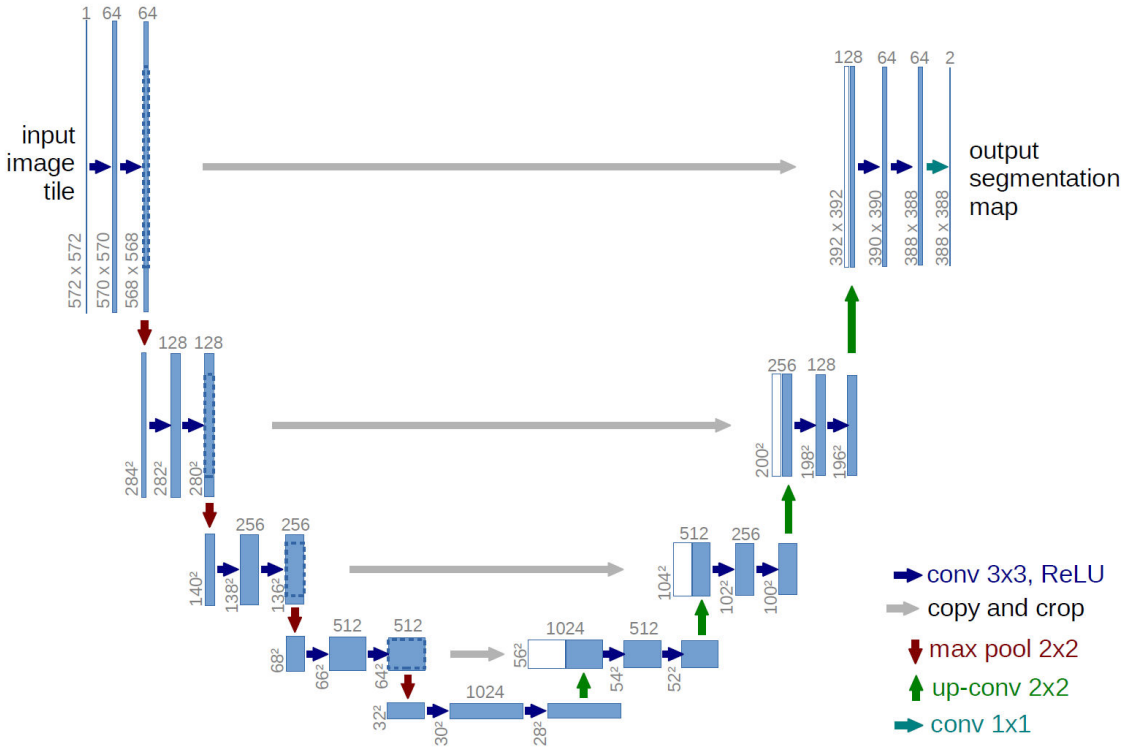


Figure 2.9: The U-Net architecture as described by Ronneberger et al. [RPB15]. We see the encoder decoder structure with skip-connections between them.

to help on our challenges while reducing the stability of the training.

In this thesis, we therefore base our network designs around the U-Net architecture.

2.2.2 Non local information combination

The need for combining local and global information appears in multiple image transformation tasks. In particular, Iizuka et al. [ISSI16] observe that colors in a photograph depend both on local features, such as an object's texture, and global context, such as being indoor or outdoor. Based on this insight, they propose a convolutional network that colorizes a gray-level picture by separately extracting global semantic features and local image features, which are later combined and processed to produce a color image. Contextual information also plays an important role in semantic segmentation, which motivates Zhao et al. [ZSQ+17] to aggregate the last layer feature maps of a classifica-

tion network in a multi-scale fashion. While we also extract local and global features separately, we exchange information between these two tracks after every layer, allowing the network to repeatedly transmit information across all image regions. Wang et al. [WGGH18] introduced a related non-local layer that mixes features between all pixels, and can be inserted at multiple points in the network to provide opportunities for non-local information exchange. While they apply more complex nonlinear mixing operations, they do not maintain an evolving global state across layers. The architecture we present in Chapter 3 has a complementary goal of aiding efficient global coordination between non-co-located points. Our scheme also opens up novel pathways, allowing information to be directly transmitted between distant image regions.

2.2.3 Multi-inputs networks

Many computer vision tasks become better posed as the number of observations increases, which calls for methods capable of handling a variable number of input images. For example, classical optimization approaches assign a data fitting error to each observation and minimize their sum. However, implementing an analogous strategy in a deep learning context remains a challenge because most neural network architectures, such as the popular U-Net used in this thesis and prior work [LDPT17, LSC18], require inputs of a fixed size and treat these inputs in an ordered manner. These architectures thus cannot simultaneously benefit from powerful learned priors as well as multiple unstructured observations. Choy et al. [CXG+16] faced this challenge in the context of multi-view 3D reconstruction and proposed a recurrent architecture that processes a sequence of images to progressively refine its prediction. However, the drawback of such an approach is that the solution still depends on the order in which the images are provided to the method – the first image has a great impact on the overall solution, while subsequent images tend to only modify details. This observation motivated Wiles et al. [WZ17] to process each image of a multi-view set through separate encoders before combining their features through max-pooling, an order-agnostic operation. Aittala et al. [AD18] and Chen et al. [CHW18] apply a similar strategy to the problems of burst image deblurring and photometric stereo, respectively. In the field of geometry processing, Qi et al. [QSMG17] also apply a pooling scheme for deep learning on point sets, and show that such an architecture is an universal approximator for functions whose inputs are set-valued. Zaheer et al. [ZKR+17] further analyze the theoretical properties of pool-

ing architectures and demonstrate superior performance over recurrent architectures on multiple tasks involving loosely-structured set-valued input data. We build on this family of work to offer a method, described in Chapter 4, that processes images captured in an arbitrary order, and that can handle un-calibrated viewing and lighting conditions.

2.3 Deep learning for material acquisition

2.3.1 Learning priors

Dror et al. [DAW01] were among the first to show that a machine learning algorithm can be trained to classify materials from low-level image features. Since then, deep learning emerged as an effective solution to related problems such as intrinsic image decomposition [NMY15, IRWM17] and reflectance and illumination estimation [RGR+17].

Given these successes, we adapt deep learning to the material acquisition problem. We use a data-driven approach to learn prior required by lightweights acquisition methods from the training data, leading to better results, simplified acquisition process and increased flexibility.

Most related to our approach is the work by Li et al. [LDPT17], who adopted an encoder-decoder architecture similar to ours to estimate diffuse reflectance and normal maps. However, their method only recovers uniform specular parameters over the material sample. In contrast, we seek to recover per-pixel specular albedo and roughness by using the cues provided by the flash in the input picture. Furthermore, they trained separate networks for different types of materials, such as wood and plastic. Rather than imposing such a hard manual clustering (which is ambiguous: consider the common case of plastic imitation of wood), we train a single all-purpose network and follow the philosophy of letting it learn by itself any special internal treatment of classes that it might find useful.

2.3.2 Rendering loss

Since our goal is to best reproduce the appearance of the captured material, we evaluate the quality of a prediction using a differentiable *rendering loss*, which compares renderings of the predicted material with renderings of the ground truth given for training and allows end to end back propagation. Rendering losses have been used by Tewari et al. [TZK+17] and Liu et al. [LCY+17] for facial capture and material editing respectively. Tewari et al. use a rendering loss to compare their reconstruction with the input image

in an unsupervised manner, while Liu et al. use it to evaluate their reconstruction with respect to both the input image and a ground-truth edited image.

For material acquisition, Aittala et al. [AAL16] also use a differentiable renderer to compare the texture statistics of their material estimates with those of an input photograph. However, they use this loss function within a standard inverse-rendering optimization rather than to train a neural network. Using deep learning, Li et al. [LDPT17] choose a L1 comparison between the predicted and ground truth BRDF parameters, which does not account for the intricate interactions between them for the final appearance of the material.

In a concurrent work to this thesis, Li et al. [LSC18] develop a method for single-image acquisition with a similar rendering loss idea. Most material acquisition methods assume a near planar surface to acquire. In further work, Li et al. [LXR+18] introduced a method for non planar surface acquisition, extending the rendering loss idea through an rendering approximation generated by a specialized deep network.

2.3.3 Multiple images acquisition

While impressive in many cases, the solutions produced by these single-image methods [DAD+18, LSC18] are largely guided by the learned priors, and often fail to reproduce important material effects simply because they are not observed in the image provided as input, or are too ambiguous to be accurately identified without additional observations. We address this limitation by designing an architecture that supports an arbitrary number of input images. In a concurrent work, Gao et al. [GLD+19] propose a solution to combine multiple picture by optimizing a deep latent vector, initializing the solution with the result of our one image network presented in Chapter 3. Their method exploits known view position and light power as well as collocated light source to correct the perspective on images and run a classical optimization comparing outputs and input pictures.

2.3.4 Synthetic data & augmentation

Training a network in a supervised manner requires a large dataset of paired input and ground truth to guide the training process. In the case of material acquisition, real world picture and SVBRDF parameters pair dataset are extremely sparse. Because the contri-

butions of shape, material and lighting are conflated in the colors of real-world pictures, many deep-learning methods for inverse rendering rely on synthetic training data to obtain the necessary supervision on these separate components [RGR+17, LDPT17, LSC18, LXR+18, LCY+17]. Given the success on previous work, we also use an entirely synthetic training dataset. While in theory image synthesis offers the means to generate an arbitrary large amount of training data, the cost of image rendering, storage and transfer limits the size of the datasets used in practice. For example, Li et al. [LSC18] report training datasets of $150{,}000$ images and for our project described in Chapter 3, we generated a dataset of $200{,}000$ images. Given the mentioned constrains, for our following project, described in Chapter 4, we develop an online data generation allowing us to provide the network with a new image at each iteration of the training, yielding up to millions of training images in practice. Our online data generation also greatly simplifies testing with different data distributions, a property that we exploit to compare multiple versions of our approach.

To augment the diversity in the training dataset, multiple techniques exist, among which most common are to apply rotation, cropping or resize during the training, to reduce the number of times the network will see the same image. A more specialized approach to materials is introduced by Li et al.[LDPT17] called *self-augmentation* to expand a small synthetic training set with semi-synthetic data based on the network's own predictions for real-world photographs. This strategy is complementary to our massive procedural data generation. In this thesis we develop an augmentation scheme for material mixing using linear combination and our online rendering process to generate a virtually infinite number of inputs pairs from a 2000 SVBRDFs dataset.

2.3.5 Resolution and scale of acquisition

Many of the methods for material acquisition based on deep learning [DAD+18, LSC18, DAD+19, GLD+19] succeed in the task by targeting flash pictures captured at a small distance from the material sample. In such configurations, the flash produces a highlight at the center of the image as well as diffuse shading on its boundary, providing information about the specular and diffuse behavior of the surface respectively, as well as complementary cues about normal variations. However, the use of a flash imposes two limitations on this family of methods. First, capturing large-scale surfaces would require the use of large, powerful flash, defeating the purpose of these lightweight methods. Sec-

ond, because flash lighting yields different visual cues in different places of the image, existing methods need to process the image in its entirety, which is problematic for deep learning methods as the networks resolution is limited by the GPU memory – related methods were typically using images of 256×256 pixel resolution. While we build on such methods to obtain our close-up SVBRDF patches, we complement them in Chapter 5 by introducing a large scale guidance image, which can be captured several meters away from the surface of interest. In addition, we assume that this large-scale image is captured under ambient lighting that varies little across the surface, so that it can be decomposed into independent tiles to fit in memory.

Our use of guidance and exemplar images makes our problem akin to *image analogies* [HJO+01], where the goal is to copy the appearance of the exemplars onto the guidance, based on a notion of similarity between exemplar and guidance pixels. The image analogies framework has been applied to a variety of problems, such as image colorization [WAM02], style transfer [FJL+16], texture transfer [DBP+15]. All these methods share the strength of providing high-level control on their output thanks to the exemplar, a feature that we now provide in the context of SVBRDF capture. Closer to our application domain is the work by Melendez et al. [MGSJW12], who used patch-based texture synthesis to transfer diffuse albedo and depth variations from small material exemplars to large façade images. However, this approach assumes that every pixel of the guidance can be put in correspondence to similar pixels of the exemplar, which yields visual artefacts when the exemplars do not contain all the material variations of the guidance image. Our deep learning approach alleviates this issue by complementing the exemplars with priors on material appearance learned from a large dataset of SVBRDFs.

Deep learning has recently been applied to several of the above image analogies problems, which inspired the design of our deep network architecture. In particular, style transfer with Adaptive Instance Normalization (AdaIN) [HB17] processes the content and style images with two separate encoders, and then transfers information about the style feature maps to the content feature maps, which are subsequently decoded to form the output. The main difference between the two approaches is that AdaIN only transfers *global statistics* of the feature maps from one encoder to the other, while we concatenate the feature maps of the two encoders to maximize information sharing. In addition, AdaIN relies on a pre-trained encoder (VGG-19), while we use dedicated encoders for the image and SVBRDF branches of our network. We show that our approach better cap-

tures the appearance of SVBRDF exemplars compared to generic style transfer. Another source of inspiration is the colorization method by Zhang et al. [ZZI+17], that allows users to provide a color histogram along with the grayscale input to control which colors should appear in the output.

The method of Chapter 5 also relates to guided super-resolution algorithms, which rely on a high-resolution guidance image to super-resolve low resolution depth or normal maps [dLDWS19, HLT16]. In particular, our deep network architecture shares ideas with the one by Hui et al. [HLT16], where features computed by the guidance encoder are concatenated to features computed by the depth map decoder. However, our problem differs since our prime goal is to augment the spatial extent of the SVBRDF exemplars rather than their resolution. As a consequence, our SVBRDF exemplars are not aligned with the guidance image, while guided super-resolution algorithms require such an alignment. In addition, we designed our method to take as input an arbitrary number of SVBRDF exemplars rather than a single depth map.

A wider overview of recent work on material acquisition methods based on deep learning from both Computer graphics and machine learning perspectives is available in the Dong [Don19] survey.

Chapter 3

Single-Image SVBRDF Capture with a Rendering-Aware Deep Network

The work presented in this chapter was done in collaboration with Miika Aittala, Fredo Durand, George Drettakis and Adrien Bousseau and published at Siggraph 2018 [DAD+ 18].

Figure 3.1: From a single flash photograph of a material sample (insets), our deep learning approach predicts a spatially-varying BRDF. See supplemental materials for animations with a moving light.

In this chapter, we introduce a deep learning method to recover spatially-varying diffuse, specular and normal maps from a single image captured under flash lighting. We achieve this goal by introducing several innovations on training data acquisition and network design.

For training, we leverage a large dataset of artist-created, procedural SVBRDFs[1] which we sample and render under multiple lighting directions. We further amplify the data by material mixing to cover a wide diversity of shading effects, which allows our network to work across many material classes.

Motivated by the observation that distant regions of a material sample often offer complementary visual cues, we design a network that combines an encoder-decoder convolutional track for local feature extraction with a fully-connected track for *global feature*

[1]Our dataset is available here: https://team.inria.fr/graphdeco/projects/deep-materials/

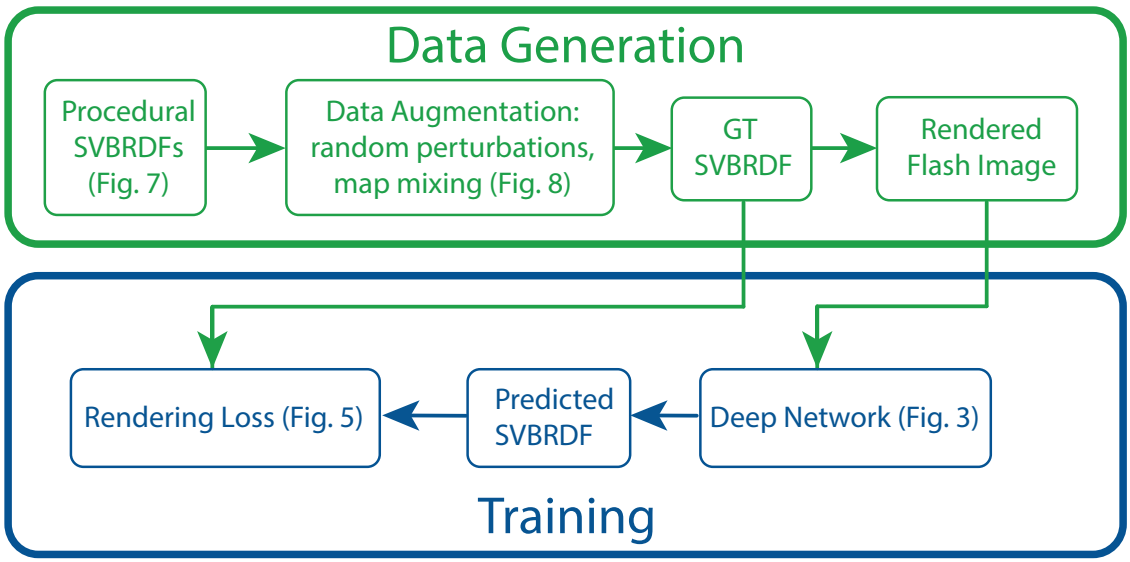


Figure 3.2: Overview of our method: we use procedural SVBRDFs to generate our ground truth (GT) training data, which we augment by random perturbations of the procedural parameters and mixing of the SVBRDF maps (Figures 3.7 and 3.8, Section 3.2). We then use physically-based rendering to synthesize the corresponding flash images. These are used to train our Deep Network (Figure 3.3, Sections 3.1.1 and 3.1.2) which compares predicted SVBRDFs and ground truth using a rendering loss (Figure 3.5, Section 3.1.3).

extraction and propagation.

Many important material effects are view-dependent, and as such ambiguous when observed in a single image. We tackle this challenge by defining the loss as a differentiable SVBRDF similarity metric that compares the *renderings* of the predicted maps against renderings of the ground truth from several lighting and viewing directions.

Combined together, these novel ingredients bring clear improvement over state of the art methods for single-shot capture of spatially varying BRDFs.

3.1 Network Architecture

Our problem boils down to translating a photograph of a material into a coinciding SVBRDF map representation, which is essentially a multi-channel image. The *U-Net* architecture [RPB15] has proven to be well suited for a wide range of similar image-to-image translation tasks [ZZI+17, IZZE17]. However, our early experiments revealed that

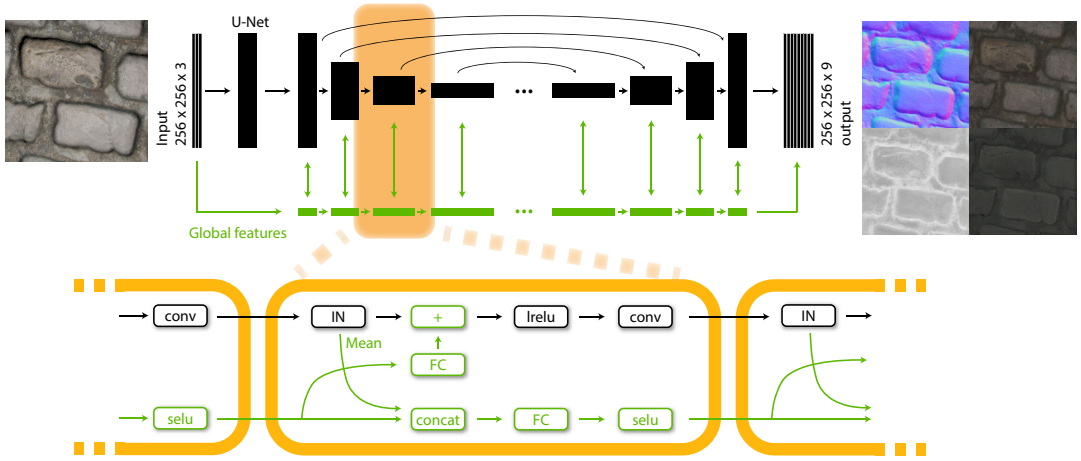


Figure 3.3: Architecture of our deep convolutional network, which takes as input a single flash-lit image (left) and predicts four maps corresponding to per-pixel normal, diffuse albedo, specular albedo and specular roughness (right). Our network follows the popular U-Net encoder-decoder architecture (black), which we complement with a new *global features* track (green) that processes vectors instead of feature maps. Taken together, the full network consists of repeating "modules", which are detailed in the bottom part of the figure. At every stage of the network, the feature means subtracted by the instance normalization after the convolutional layer are concatenated with the global feature vector, which is then processed by a fully connected layer and a non-linearity before being added to the feature maps of the next stage. *IN* and *FC* denote instance normalizations and fully connected layers respectively. We use SELU [KUMH17] and leaky ReLu activation functions. In the decoder, the set of layers also includes a skip-connection concatenation and a second convolution, which we omit for clarity. We provide the code of our network to allow reproduction.

despite its multi-scale design, this architecture remains challenged by tasks requiring the fusion of distant visual information. We address this limitation by complementing the U-Net with a parallel *global features* network tailored to capture and propagate global information.

3.1.1 U-Net Image-to-Image Network

We adopt the U-Net architecture as the basis of our network design, and follow Isola et al. [IZZE17] for most implementation details. Note however that we do not use their *discriminator* network, as we did not find it to yield a discernible benefit in our problem.

We now briefly describe the network design. We provide the code of our network and its learned weights to allow reproduction of our results[2].

As illustrated in Figure 3.3, our base network takes a 3-channel photograph as input and outputs a 9-channel image of SVBRDF parameters – 3 channels for the RGB diffuse albedo, 3 channels for the RGB specular albedo, 2 channels for the x and y components of the normal vector in tangent plane parameterization, and 1 channel for the specular roughness. We use low dynamic range images as input photographs due to the ease of acquisition, and let the network learn how to interpret the saturated highlight regions. Regardless, the dynamic range of flash photographs can still be large. We flatten the dynamic range by transforming the input image into logarithmic space and compacting it to the range $[0, 1]$ via the formula $\frac{\log(x+0.01)-\log 0.01}{\log(1.01)-\log(0.01)}$.

The input image is processed through a sequence of 8 convolutional layers that perform downsampling (the encoder), followed by a sequence of 8 upsampling and convolutional layers (the decoder). Such a hourglass-shaped network gradually reduces the resolution of the image while increasing the feature size, forcing the encoder to compress the relevant information into a concise, global feature vector. The task of the decoder is to expand these global features back into a full-sized image that matches the training target. However, while the bottleneck is critical to aggregate spatially-distant information, it hinders the reproduction of fine details in the output. Following Ronneberger et al. [RPB15], we mitigate this issue by introducing skip connections between same-sized layers of the encoder and decoder, helping the decoder to synthesize details aligned with the input at each spatial scale.

Prior to the decoder, we insert a single convolutional layer with 64 output feature channels. The feature counts in the encoder downscaling layers are 128, 256, 512, 512, 512, 512, 512 and 512. The downsampling is implemented by using a stride of 2 in the convolutions. In the decoder, the same feature counts are used in reverse order. At each scale, a nearest-neighbor upsampling is followed by concatenation of encoder features, and two convolutions. We use the filter size $[4, 4]$ across all layers. For nonlinearities we use the leaky ReLu activation function with a weight 0.2 for the negative part. The final output is mapped through a sigmoid to enforce output values in the range $[0, 1]$.

Following each convolution layer (or pair thereof), we apply instance normalization,

[2] https://team.inria.fr/graphdeco/projects/deep-materials/

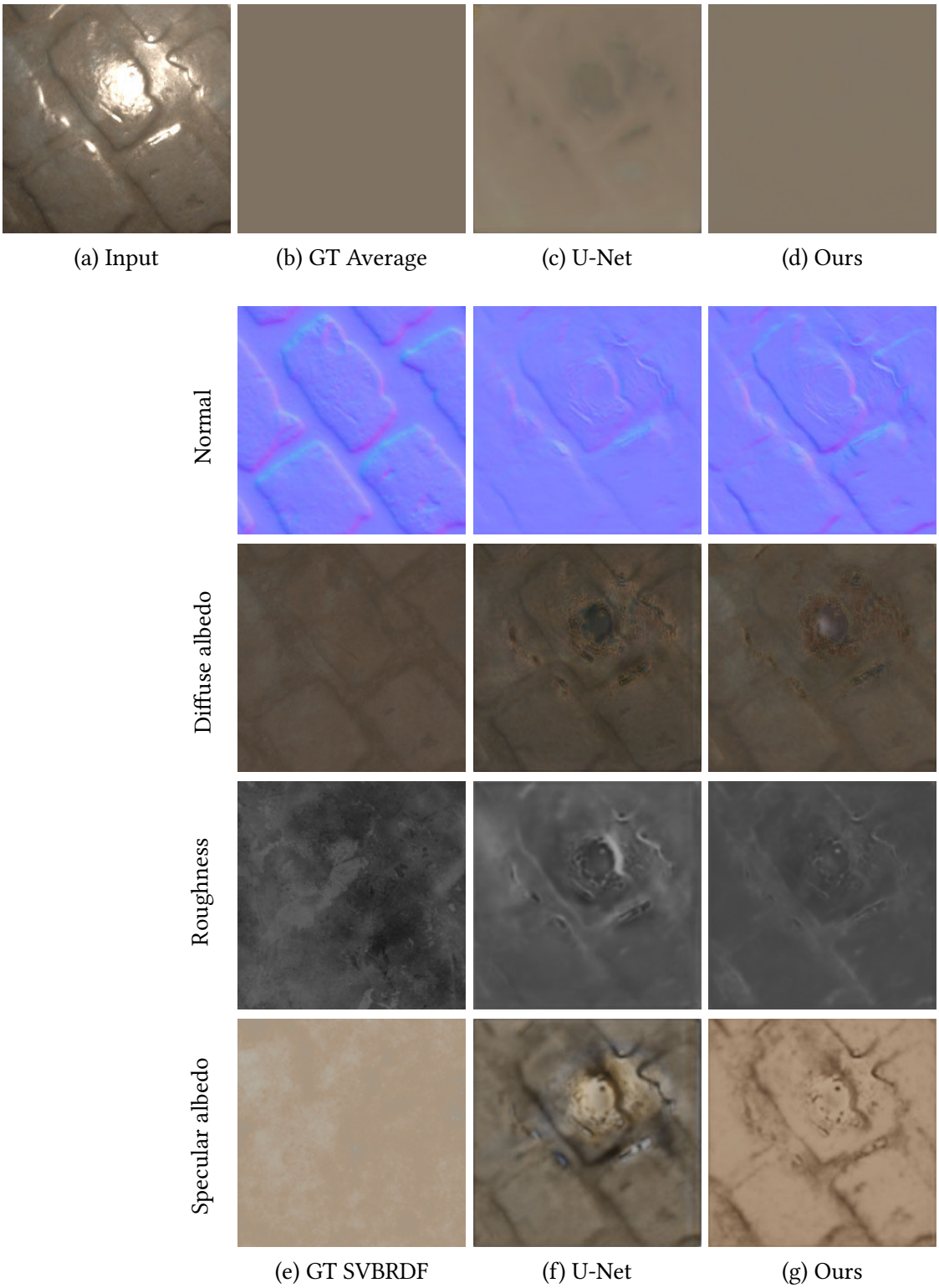
which stabilizes training on image generation tasks [UVL17, IZZE17]. Finally, we regularize by applying dropout at 50% probability on the three coarsest layers of the decoder.

3.1.2 Global Features Network

Distant regions of a material sample often offer complementary information to each other for SVBRDF recovery. This observation is at the heart of many past methods for material capture, such as the work of Lensch et al. [LKG+03] where the SVBRDF is assumed to be spanned by a small set of basis BRDFs, or the more recent work of Aittala et al. [AWL15, AAL16] where spatial repetitions in the material sample are seen as multiple observations of a similar SVBRDF patch. Taking inspiration from these successful heuristics, we aim for a network architecture capable of leveraging redundancies present in the data.

The hourglass shape of the U-Net results in large footprints of the convolution kernels at coarse spatial scales, which in theory provide long-distance dependencies between output pixels. Unfortunately, we found that this multi-scale design is not sufficient to properly fuse information for our problem. We first illustrate this issue on a toy example, where we trained a network to output an image of the average color of the input, as shown in Figure 3.4 (top row). Surprisingly, the vanilla U-Net performs poorly on this simple task, failing to output a constant-valued image. A similar behavior occurs on our more complex task, where visible residuals of the specular highlight and other fine details pollute the output maps where they should be uniform (Figure 3.4, 2nd to 4th row).

In addition, we hypothesize that the ability of the network to compute global information is partly hindered by instance (or batch) normalization, which standardizes the learned features after every convolutional layer by enforcing a mean and standard deviation learned from training data. In other words, while the normalization is necessary to stabilize training, it actively counters the network's efforts to maintain non-local information about the input image. In fact, instance normalization has been reported to improve artistic style transfer because it eliminates the output's dependence on the input image contrast [UVL17]. This is the opposite of what we want. Unfortunately, while we tried to train a U-Net without normalization, or with a variant of instance normalization without mean subtraction, these networks yielded significant residual shading in all maps.



(a) Input (b) GT Average (c) U-Net (d) Ours

(e) GT SVBRDF (f) U-Net (g) Ours

Figure 3.4: We trained a U-Net convolutional network to predict an image of the average color of the input (top row). Surprisingly, the basic U-Net fails to produce a constant image (c). Similar artifacts appear when using the U-Net for SVBRDF prediction (f). We address this issue by complementing the U-Net with a parallel network that explicitly computes and propagates global features. This approach succeeds in computing the average image (d) and reduces artifacts in SVBRDF maps (g).

We propose a network architecture that simultaneously addresses both of these shortcomings. We add a parallel network track alongside the U-Net, which deals with *global* feature vectors instead of 2D feature maps. The structure of this global track mirrors that of the main convolutional track, with convolutions changed to fully connected layers and skip connections dropped, and with identical numbers of features. See Figure 3.3 for an illustration and details of this architecture. The global and convolutional tracks exchange information after every layer as follows:

- Information from the convolutional track flows to the global track via the instance normalization layers. Whereas the standard procedure is to discard the means that are subtracted off the feature maps by instance normalization, we instead incorporate them into the global feature vector using concatenation followed by a fully connected layer and a nonlinearity. For the nonlinearity, we use the Scaled Exponential Linear Unit (SELU) activation function, which is designed to stabilize training for fully connected networks [KUMH17].

- Information from the global track is injected back into the local track after every convolution, but before the nonlinearity. To do so, we first transform the global features by a fully connected layer, and add them onto each feature map like biases.

Our global feature network does not merely preserve the mean signal of a given feature map – it concatenates the means to form a global feature vector that is processed by fully connected layers before being re-injected in the U-Net at multiple scales. Each pair of these information exchanges forms a nonlinear dependency between every pixel, providing the network with means to arrive at a consistent solution by repeatedly transmitting local findings between different regions. In particular, the common case of near-constant reflectance maps becomes easier for the network to express, as it can source the constant base level from the global features and the fine details from the convolutional maps (Figure 3.4).

3.1.3 Rendering Loss

Our network outputs a set of maps that describe BRDF parameters, such as specular roughness and albedo, at every surface point. The choice of parameterization is arbitrary, as it merely acts as a convenient proxy for the actual object of interest: the spatio-angular

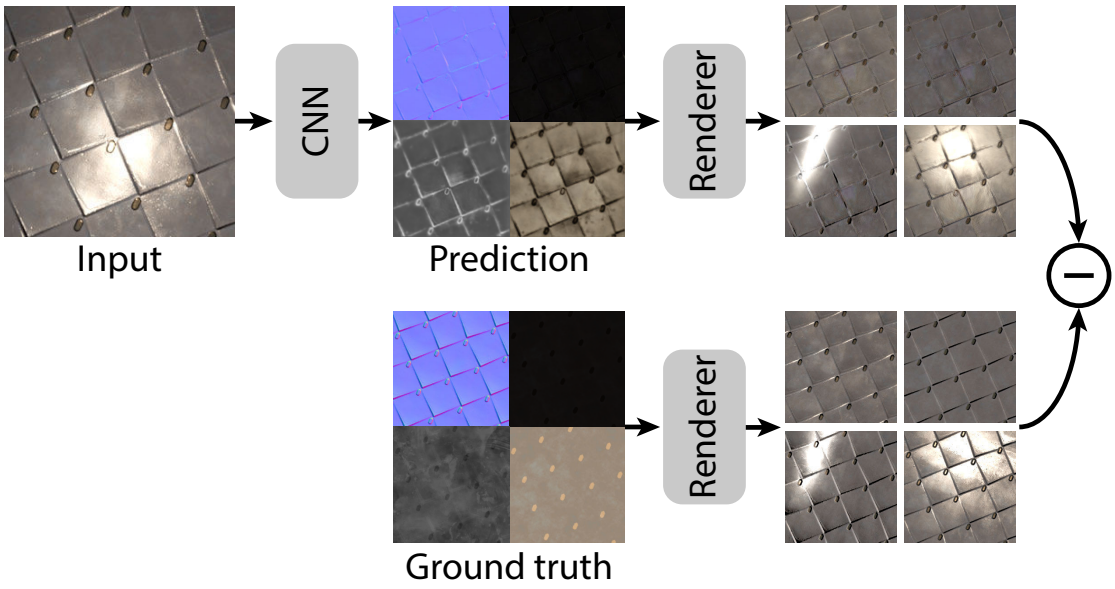
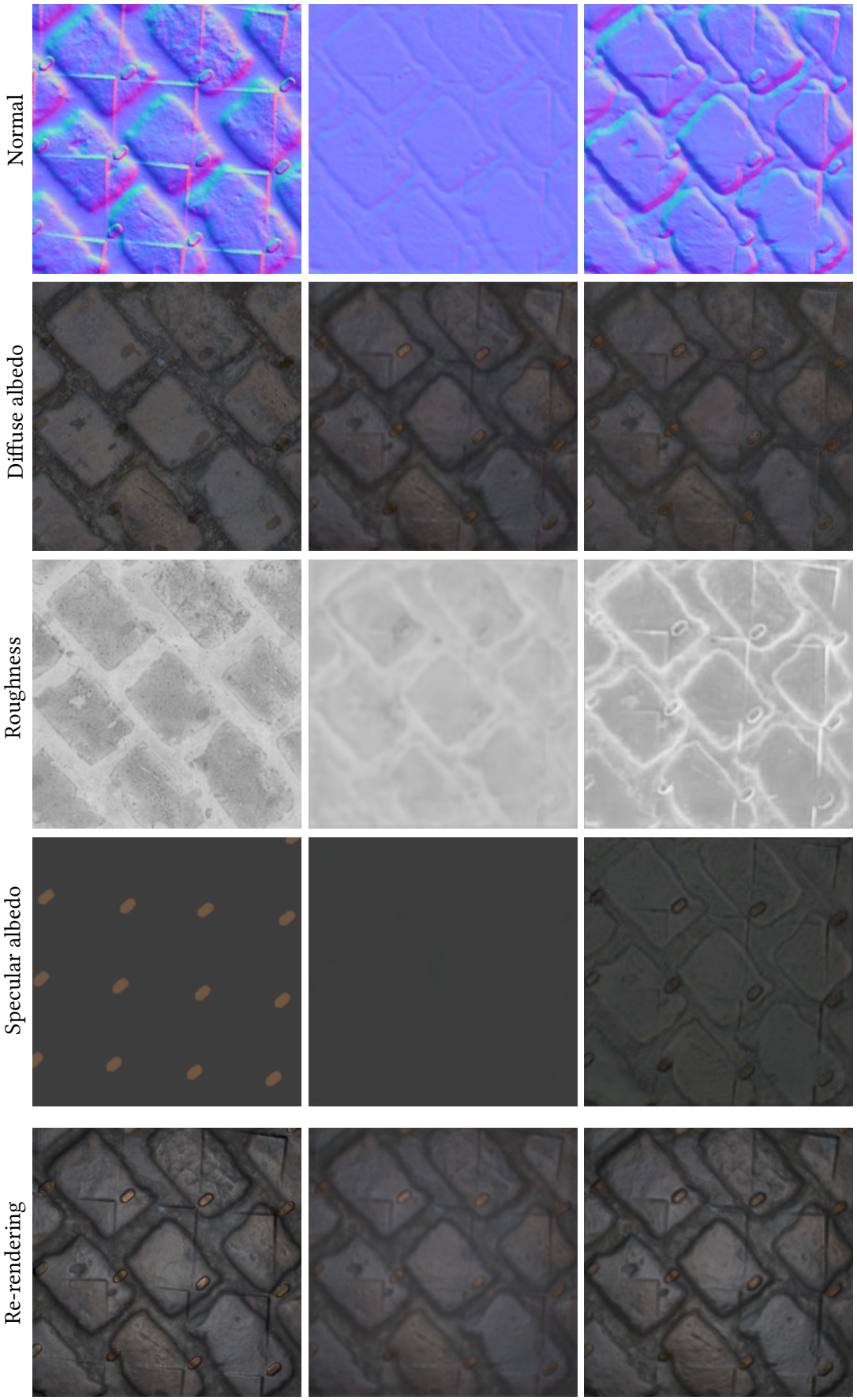


Figure 3.5: Our rendering loss compares the appearance of the predicted SVBRDF and ground truth by rendering both under the same random lighting and viewing configurations.

appearance of the SVBRDF. In fact, the parameterizations of popular BRDF models arise from a combination of mathematical convenience and relative intuitiveness for artists, and the numerical difference between the parameter values of two (SV)BRDFs is only weakly indicative of their visual similarity.

We propose a loss function that is *independent* of the parameterization of either the predicted or the target SVBRDF, and instead compares their *rendered appearance*. Specifically, any time the loss is evaluated, both the ground truth SVBRDF and the predicted SVBRDF are rendered under identical illumination and viewing conditions, and the resulting images are compared pixel-wise. We use the same Cook-Torrance BRDF model [CT82] for the ground truth and prediction, but our loss function could equally be used with representations that differ between these two quantities.

We implement the rendering loss using an in-network renderer, similarly to Aittala et al. [AAL16]. This strategy has the benefits of seamless integration with the neural network training, automatically-computed derivatives, and automatic GPU acceleration. Even complicated shading models are easily expressed in modern deep learning frameworks



(a) GT SVBRDF (b) l_1 loss (c) Rendering loss

Figure 3.6: When trained with the l_1 loss (b), the SVBRDF predicted by the network for a test input image does not accurately reproduce the appearance of the target material when rendered. A network trained using the rendering loss (c) produces an SVBRDF

such as TensorFlow [AAB+15]. In practice, our renderer acts as a pixel shader that evaluates the rendering equation at each pixel of the SVBRDF, given a pair of view and light directions (Figure 3.5). Note that this process is performed in the SVBRDF coordinate space, which does not require to output pixels according to the perspective projection of the plane in camera space.

Using a fixed finite set of viewing and lighting directions would make the loss blind to much of the angular space. Instead, we formulate the loss as the average error over *all* angles, and follow the common strategy of evaluating it stochastically by choosing the angles at random for every training sample, in the spirit of stochastic gradient descent. To ensure good coverage of typical conditions, we use two sets of lighting and viewing configurations:

- The first set of configurations is made of orthographic viewing and lighting directions, sampled independently of one another from the cosine-weighted distribution over the upper hemisphere. The cosine weighting assigns a lower weight to grazing angles, which are observed less often in images due to foreshortening.

- While the above configurations cover all angles in theory, in practice it is very unlikely to obtain mirror configurations, which are responsible for visible highlights. Yet, highlights carry rich visual information about material appearance, and should thus contribute to the SVBRDF metric. We ensure the presence of highlights by introducing mirror configurations, where we only sample the lighting direction from the cosine distribution, and use its mirror direction for the viewing direction. We place the origin at a random position on the material plane, and choose independent random distances for both the light and the camera according to the formula $\exp(d)$, where $d \sim \mathrm{Normal}(\mu = 0.5, \sigma = 0.75)$ for a material plane of size 2×2. The net effect of these configurations is to produce randomly-sized specular highlights at random positions.

We compare the logarithmic values of the renderings using the l_1 norm. The logarithm is used to control the potentially extreme dynamic range of specular peaks, and because we are more concerned with relative than absolute errors. To reduce the variance of the stochastic estimate, for every training sample we make 3 renderings in the first config-

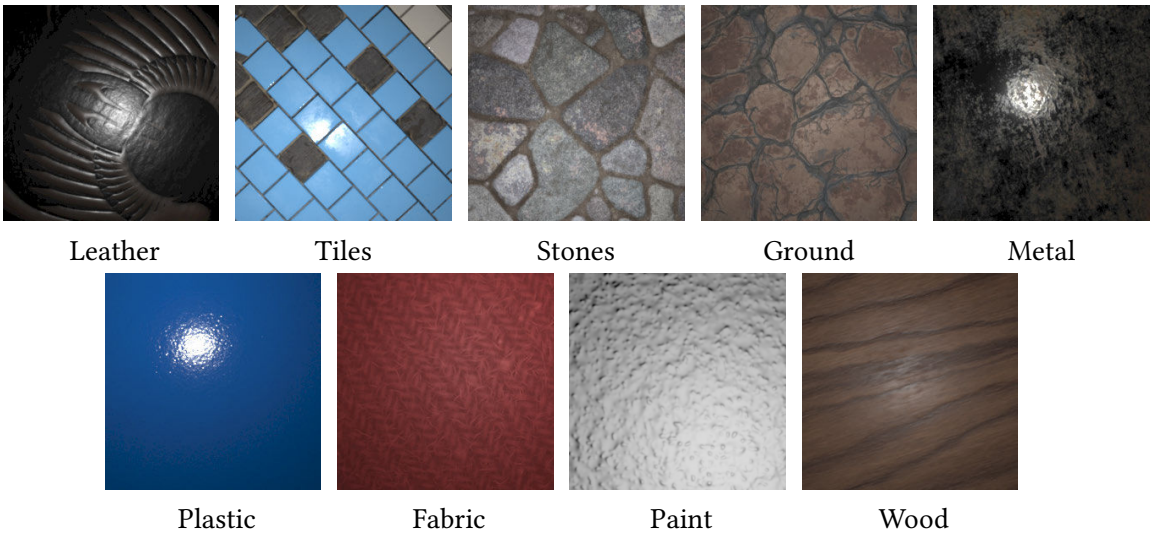


Figure 3.7: Example parametric SVBRDFs for each original material class. We produce our final training set by perturbing and mixing such SVBRDFs.

uration and 6 renderings in the second, and average the loss over them. We provide a detailed pseudo-code of our rendering loss in Appendix A.1.

Figure 3.6 compares the output of our network when trained with a naive l_1 loss against the output obtained with our rendering loss. While the l_1 loss produces plausible maps when considered in isolation, these maps do not reproduce the appearance of the ground truth once re-rendered. In contrast, the rendering loss yields a more faithful reproduction of the ground truth appearance.

3.1.4 Training

We train the network with batch size of 8 for $400{,}000$ iterations, using the Adam optimization algorithm [KB15] with a fixed learning rate of 0.00002. The training takes approximately one week on a TitanX GPU.

3.2 Procedural Synthesis of Training Data

While several recent papers have shown the potential of synthetic data to train neural networks [SQLG15, ZSY+17, RVRK16], care must be taken to generate data that is representative of the diversity of real-world materials we want to capture. We address this challenge by leveraging Allegorithmic Substance Share [All19b], a dataset of more than

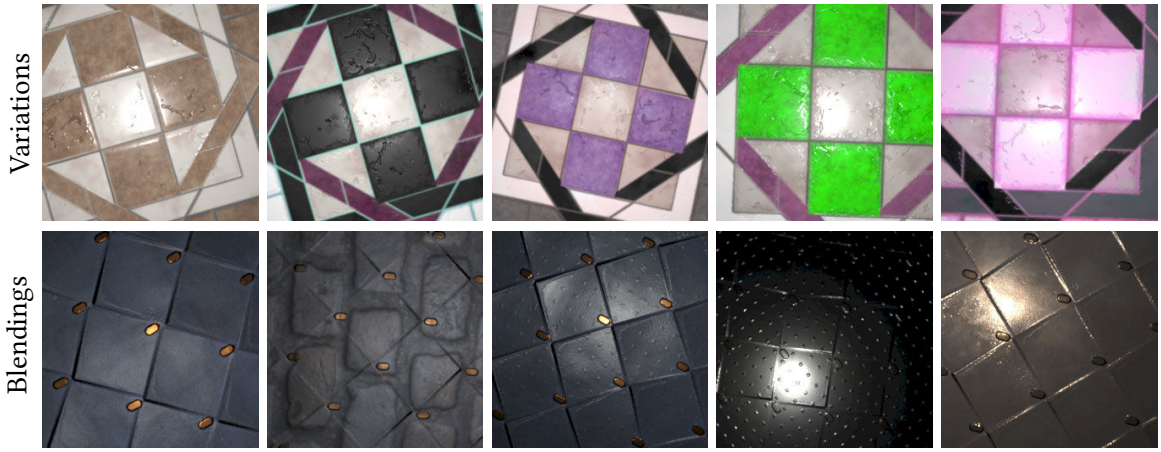
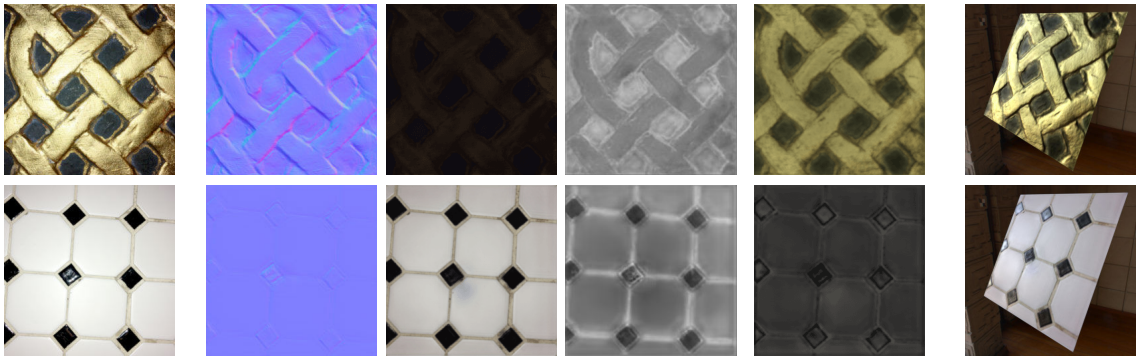


Figure 3.8: Data augmentation. We create variations of each parametric SVBRDF by randomly perturbing its parameters (first row). We additionally augment our dataset by blending pairs of SVBRDFs (second row). Finally, we render each SVBRDF under various orientations, scaling and lighting conditions (both rows).



Input Normal Diffuse albedo Roughness Specular albedo Re-rendering

Figure 3.9: Based on the input photographs (left), our method has recovered a set of SVBRDF maps that exhibit strong spatially varying specular roughness and albedo effects. The gold-colored paint (top) and the highly glossy black tiles (bottom) are clearly visible in the re-renderings of SVBRDF under environment illumination (right).

800 procedural SVBRDFs designed by a community of artists from the movie and video game industry. This dataset has several key features relevant to our needs. First, it is representative of the materials artists care about. Second, each SVBRDF is rated by the community, allowing us to select the best ones. Third, each SVBRDF exposes a range of procedural parameters, allowing us to generate variants of them for data augmentation. Finally, each SVBRDF can be converted to the four Cook-Torrance parameter maps we want to predict [CT82].

We first curated a set of 155 high-quality procedural SVBRDFs from 9 material classes – paint (6), plastic (5), leather (13), metal (35), wood (23), fabric (6), stone (25), ceramic tiles (29), ground (13), some of which are illustrated in Figure 3.7. We also selected 12 challenging procedural SVBRDFs (6 metals, 3 plastics, 3 woods) to serve as an independent testing set in our comparison to Li et al. [LDPT17]. Together with two artists, we identified the procedural parameters that most influence the appearance of each of our training SVBRDFs. We obtained between 1 and 36 parameters per SVBRDF (7 on average), for which we manually defined the valid range and default values.

We then performed four types of data augmentation. First, we generated around 1,850 variants of the selected SVBRDFs by applying random perturbations to their important parameters, as illustrated in Figure 3.8 (top). Second, we generated around 20,000 convex combinations of random pairs of SVBRDFs, which we obtained by α-blending their maps. The mixing greatly increases the diversity of low-level shading effects in the training data, while staying close to the set of plausible real-world materials, as shown in Figure 3.8 (bottom). Third, we rendered each SVBRDF 10 times with random lighting, scaling and orientation. Finally, we apply a random crop on each image at training time, so that the network sees slightly different data at each epoch.

The scene we used to render each SVBRDF is composed of a textured plane seen from a fronto-parallel camera and dimensioned to cover the entire image after projection. The light is a small white emitting sphere positioned in a plane parallel to the material sample, at a random offset from the camera center. The camera has a field of view of 50° to match the typical field of view of cell-phone cameras after cropping to a square, and is positioned at a fixed distance from the material sample. Note that there is a general ambiguity between the scale of the SVBRDF, the distance of the camera, and the strength of the light, which is why we hold the latter parameters fixed. However, since such

parameters are unknown in our casual capture scenario, the albedo maps we obtain from real pictures at test time are subject to an arbitrary, global scale factor.

We used the Mitsuba renderer [Jak10], for which we implemented the Cook-Torrance BRDF model [CT82] with GGX normal distribution [WMLT07] to match the model used in Allegorithmic Substance. We rendered each SVBRDF as a linear low-dynamic range image, similar to gamma-inverted photographs captured with a cell-phone. We also used Mitsuba to render the parameter maps after random scaling and rotation of the material sample, which ensures that the maps are aligned with the material rendering and that the normal map is expressed in screen coordinate space rather than texture coordinate space. Our entire dataset of around $200{,}000$ SVBRDFs took around 16 hours to generate on a cluster of 40 CPUs.

Figure 3.10: A selection of results from our method on real-world photographs. In each image pair, the left image is a photograph of a surface, and the right image is a re-rendering of the SVBRDF inferred from that image. The illumination environment in the re-renderings is an interior space with a large window on the left.

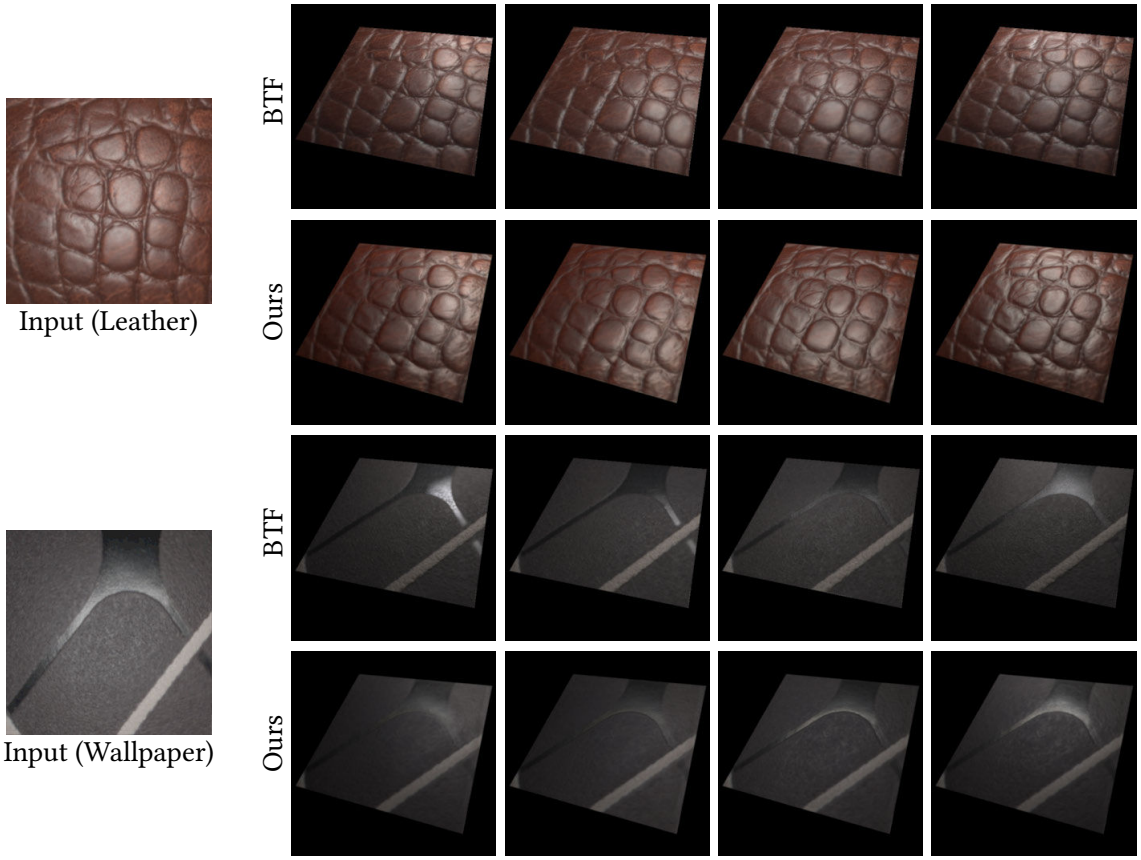


Figure 3.11: Comparison between relighting of our predictions and of measured BTFs [WGK14].

Figure 3.12: Comparison between relighting of our prediction and real pictures under approximately the same lighting configurations. We adjusted the white balance of the results to best match the one of the input.

3.3 Evaluation

We now evaluate our approach on real-world photographs and compare it with recent methods for single-image SVBRDF capture. We refer the reader to the supplemental materials for more results and animated visualisations.

3.3.1 Real-world photographs

We used regular cell phones (iPhone SE and Nexus 5X) and their built-in flash units to capture a dataset of nearly 350 materials on which we applied our method. We cropped the images to approximate the field of view used in the training data. The dataset includes samples from a large variety of materials found in domestic, office and public interiors, as well as outdoors. In fact, most of the photographs were shot during a casual walk-around within the space of a few hours.

Figures 3.1 and 3.10 show a selection of representative pairs of input photographs, and corresponding re-renderings of the results under novel environment illumination. The results demonstrate that the method successfully reproduces a rich set of reflectance effects for metals, plastics, paint, wood and various more exotic substances, often mixed together in the same image. We found it to perform particularly well on materials exhibiting bold large-scale features, where the normal maps capture sharp and complex geometric shapes from the photographed surfaces.

Figure 3.9 shows our result for two materials with interesting spatially varying specularity behavior. The method has successfully identified the gold paint in the specular albedo map, and the different roughness levels of the black and white tiles. The latter feature shows good consistency across the spatially distant black squares, and we find it particularly impressive that the low roughness level was apparently resolved based on the small highlight cues on the center tile and the edges of the outer tiles. For most materials, the specular albedo is resolved as monochrome, as it should be. Similar globally consistent behavior can be seen across the result set: cues from sparsely observed specular highlights often inform the specularity across the entire material.

Note that our dataset contains several duplicates, i.e. multiple shots of the same material taken from slightly different positions. Their respective SVBRDF solutions generally show good consistency among each other. We also captured a few pictures with an SLR camera, for which the flash is located further away from the lens than cell phones.

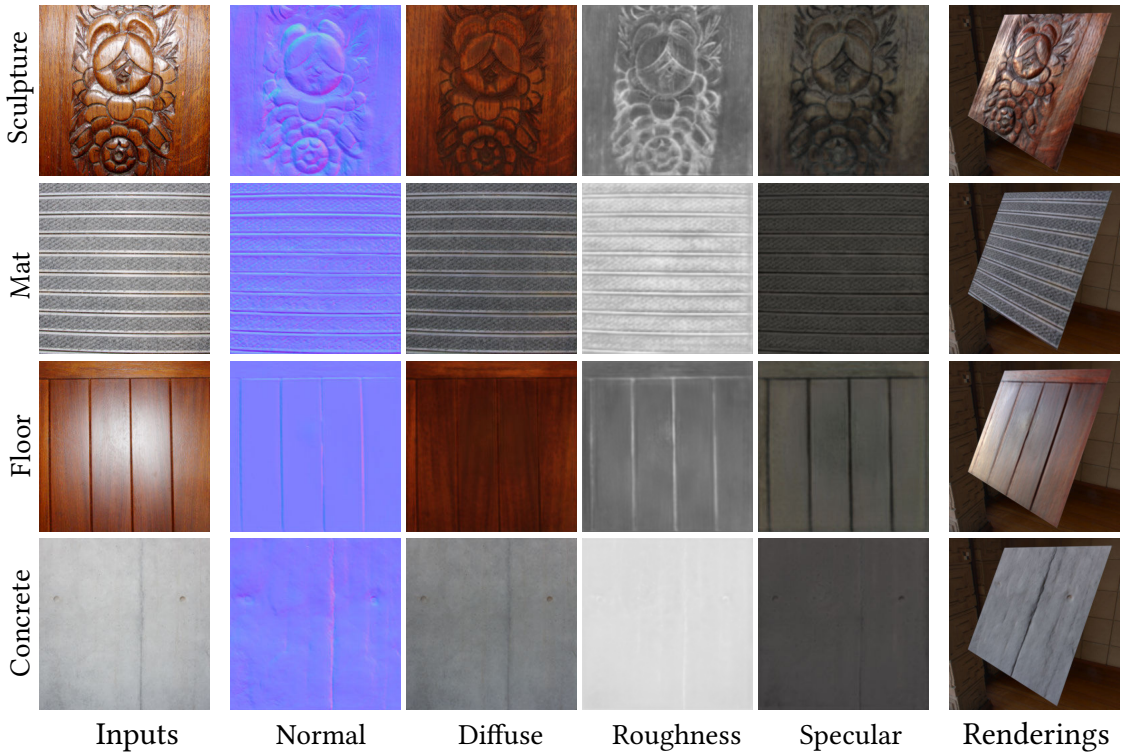


Figure 3.13: Results of our method with a picture taken using a SLR and its flash.

Figure 3.13 presents the resulting predicted maps, showing that our method is robust to varying positions of the flash.

3.3.2 Comparisons

3.3.2.1 Relighting

Figure 3.11 provides a qualitative comparison between renderings of our predictions and renderings of measured Bidirectional Texture Functions (BTFs) [WGK14] under the same lighting conditions. While BTFs are not parameterized according to the 4 maps we estimate, they capture ground-truth appearance from arbitrary view and lighting conditions, which ultimately is the quantity we wish to reproduce. Our method provides a faithful reproduction of the appearance of the leather. It also captures well the spatially-varying specularity of the wallpaper, even though it produces slightly more blurry highlights. Please refer to supplemental materials for additional results on 20 BTFs.

In addition, Figure 3.12 compares renderings of our predictions with real photographs

under approximately similar lighting conditions. Our method is especially effective at capturing the normal variations of this wood carving.

3.3.2.2 Aittala et al. [AWL15, AAL16]

The method by Aittala et al. [AAL16] is the most related to ours in terms of input, since it also computes an SVBRDF representation from a single flash-lit photograph. However, Aittala et al. [AAL16] exploit redundancy in the input picture by assuming that the material is *stationary*, *i.e.* consists of small textural features that repeat throughout the image.

We compare our method to theirs by feeding photographs from their dataset to our network (Figure 3.14). Despite the similar input, the two approaches produce different outputs: whereas we produce a map that represents the entire input photo downsampled to 256×256, their method produces a tile that represents a small piece of the texture at high resolution. Furthermore, the BRDF models used by the methods are different. To aid comparison, we show re-renderings of the material predicted by each method under identical novel lighting conditions.

Both methods produce a good result, but show a clearly different character. The method of Aittala et al. [AAL16] recovers sharp textural details that are by construction similar across the image. For the same reason, their solution cannot express larger-scale variations, and the result is somewhat repetitive. In contrast, our solution shows more interesting large-scale variations across the image, but lacks some detail and consistency in the local features.

Most of our real-world test images violate the stationarity requirement, and as such would not be suitable for the method of Aittala et al. [AAL16]. Our method also has the advantage in speed: whereas Aittala et al. [AAL16] use an iterative optimization that takes more than an hour per material sample, our feedforward network evaluation is practically instant.

Figure 3.14 also contains results obtained with an earlier method by Aittala et al. [AWL15]. This method also assumes stationary materials, and requires an additional no-flash picture to identify repetitive details and their large-scale variations. Our approach produces similar results from a single image, although at a lower resolution.

Table 3.1: RMSE comparison between Li et al. [LDPT17] and our method. Due to the use of different parametrizations, we cannot compute RMSE on specular terms for Li et al. [LDPT17]. As their output albedo maps can have a different scaling than the ground truth with respect to lighting, we evaluate the re-rendering and diffuse albedo RMSE with multiple scaling factors on the albedo, and keep the best one (0.27).

Method	Li et al.	Ours
Re-Rendering error	0.169	0.083
Normal error	0.046	0.035
Diffuse albedo error	0.090	0.019
Specular albedo error	NA	0.050
Specular roughness error	NA	0.129

3.3.2.3 Li et al. [LDPT17]

The method by Li et al. [LDPT17] is based on a similar U-Net convolutional network as ours. However, it has been designed to process pictures captured under environment lighting rather than flash lighting, and it predicts a constant specular albedo and roughness instead of spatially-varying maps. We first compare the two methods on our synthetic test set for which we have the ground truth SVBRDFs (Figure 3.16 and Table 3.1). For a fair comparison, we tested the method by Li et al. on several renderings of the ground truth, using different environment maps and different orientations. We then selected the input image that gave the best outcome. We compare the results of the two methods qualitatively with re-renderings under a mixed illumination composed of an environment map enriched with a flash light, so as to ensure that neither method has an advantage. For quantitative comparison, we compute the RMSE of each individual map, as well as the RMSE of re-renderings averaged over multiple point lighting conditions; our results have systematically lower error.

Overall, our method reproduces the specularity of the ground truth more accurately, as evidenced by the sharpness of reflections and highlights in the re-renderings. We believe this is due to our use of near-field flash illumination, as the apparent size and intensity of the highlight caused by the flash is strongly indicative of the overall glossiness and albedo levels. The method of Li et al. [LDPT17] must rely on more indirect and ambiguous cues to make these inferences. While such cues are available in the input images – for example, the reflections of the illumination environment are blurred to different

degrees – their method has not reached an equally accurate estimate of the specular roughness.

Similarly, flash illumination highlights the surface normal variations by introducing spatially varying directional shading effects into the image. Such variations do also have a characteristic appearance in environment-lit images, but interpreting these cues may be more difficult due to ambiguities and uncertainties related to the unknown lighting environment. Consequently, the normal maps recovered by Li et al. [LDPT17] are also less accurate than ours.

We then compare the two methods on real pictures, captured with a flash for our approach and without for the approach by Li et al. (Figure 3.17). Overall, the relative performance of the methods appears similar to the synthetic case.

3.3.3 Limitations

Despite the diversity of results shown, the architecture of our deep network imposes some limitations on the type of images and materials we can handle.

In terms of input, our network processes images of 256×256 pixels, which prevents it from recovering very fine details. While increasing the resolution of the input is an option, it would increase the memory consumption of the network and may hinder its convergence. Recent work on iterative, coarse-to-fine neural image synthesis represents a promising direction to scale our approach to high-resolution inputs [CK17, KALL18]. Our network is also limited by the low dynamic range of input images. In particular, sharp, saturated highlights sometimes produce residual artifacts in the predicted maps as the network struggles to inpaint them with plausible patterns (Figure 3.15). We also noticed that our network tends to produce correlated structures in the different maps. As a result, it fails on materials like the one in Figure 3.15 (top row), where the packaging has a clear coat on top of a textured diffuse material. This behavior may be due to the fact that most of the artist-designed materials we used for training exhibit correlated maps. Finally, while our diverse results show that our network is capable of exploiting subtle shading cues to infer SVBRDFs, we observed that it resorts to naive heuristics in the absence of such cues. For example, the normal map for the wool knitting in Figure 3.15 suggests a simple "dark is deep" prior.

In terms of output, our network parameterizes an SVBRDF with four maps. Additional

maps should be added to handle a wider range of effects, such as anisotropic specular reflections. The Cook-Torrance BRDF model we use is also not suitable for materials like thick fabric or skin, which are dominated by multiple scattering. Extending our approach to such materials would require a parametric model of their spatially-varying appearance, as well as a fast renderer to compute the loss. Finally, since our method only takes a fronto-parallel picture as input, it never observes the material sample at grazing angle, and as such cannot recover accurate Fresnel effects.

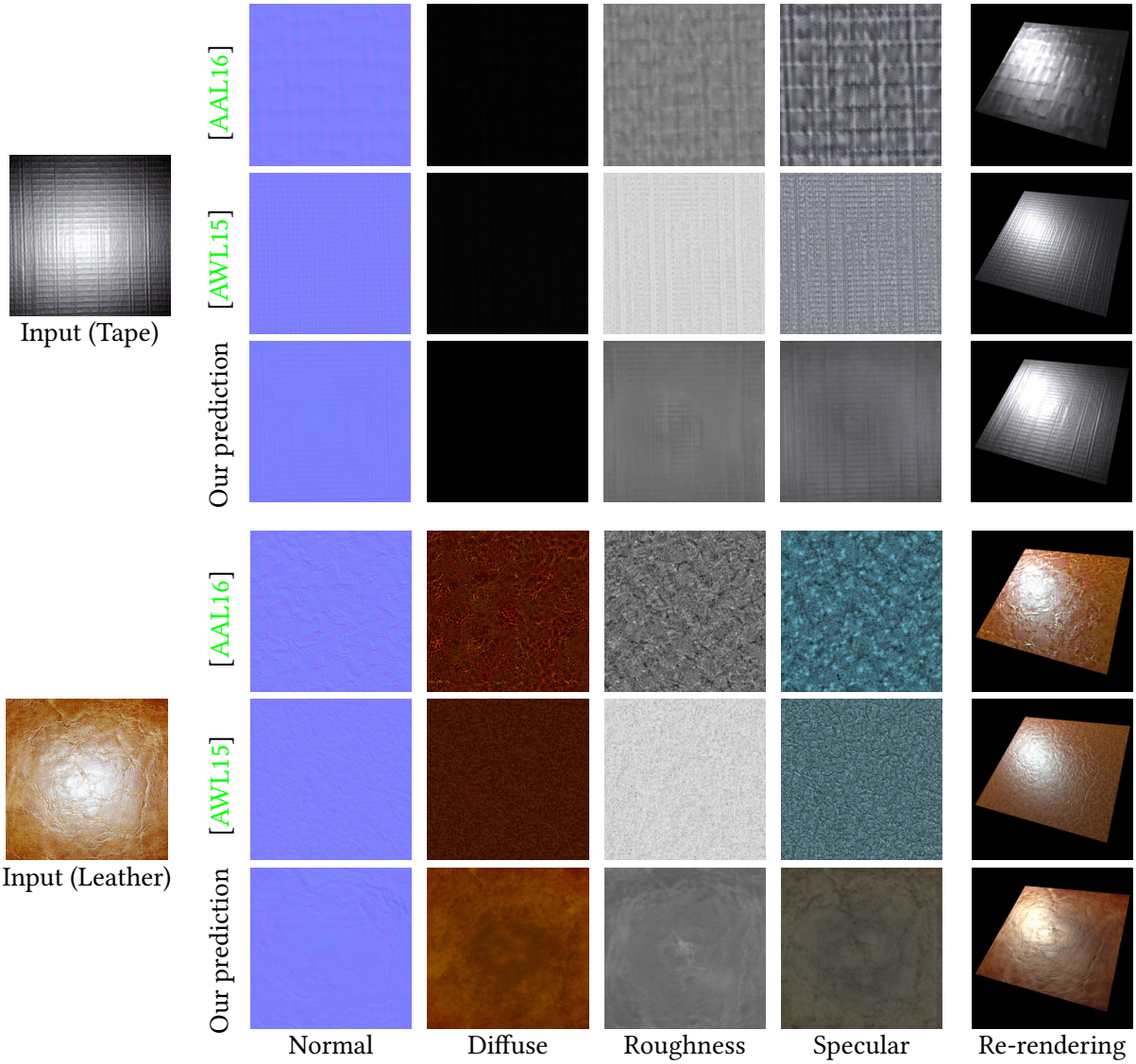
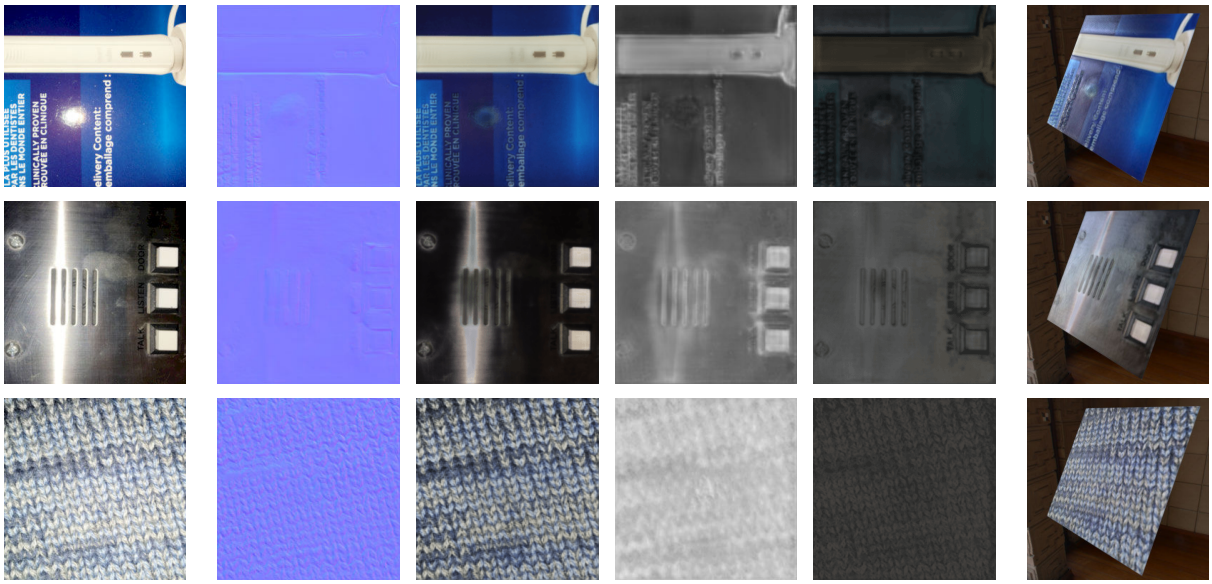


Figure 3.14: Comparison with Aittala et al. [AWL15, AAL16]. Note that the maps (other than the normals) are not directly comparable due to different parametrization of the BRDF models. The solution of Aittala et al. [AAL16] corresponds to a small region of about 15% of the image dimension, intended to be repeated by texture synthesis or tiling. The earlier method by Aittala et al. [AWL15] captures the entire input image but requires an additional no-flash picture for guidance. In contrast, our method reproduces the large-scale features well, and is applicable to non-repetitive materials captured with a single flash picture.



Input Normal Diffuse Roughness Specular Re-rendering

Figure 3.15: Failure cases and performance on materials violating our assumptions. Our method generally struggles with otherwise uniform surfaces exhibiting structured albedo detail, such as the text and the photograph on the product packaging (top). Highly concentrated specular highlights are sometimes missed and result in overestimated roughness and occasional highlight removal artifacts (top). Materials outside the scope of the training data (e.g. anisotropic brushed metal, center) cannot be reproduced properly, and result in an undefined assignment of the apparent shading effects (the streak of the specular highlight) into the various maps of the SVBRDF. Nevertheless, the method can produce reasonable approximations for materials violating the assumptions, with varying degrees of success, as seen on the fuzzy wool (bottom).

3.4 Conclusion

In this chapter we present an architecture able to extract SVBRDF parameters from a single flash picture. We leverage computer graphics knowledge to design a rendering loss and create a synthetic dataset and we show that our training on synthetic data generalizes well to real world pictures. While our method generates convincing results for most input pictures we tried, one image is sometime not enough to disambiguate all the materials properties. We address this limitation in Chapter 4 by allowing to aggregate the information available in multiple photographs.

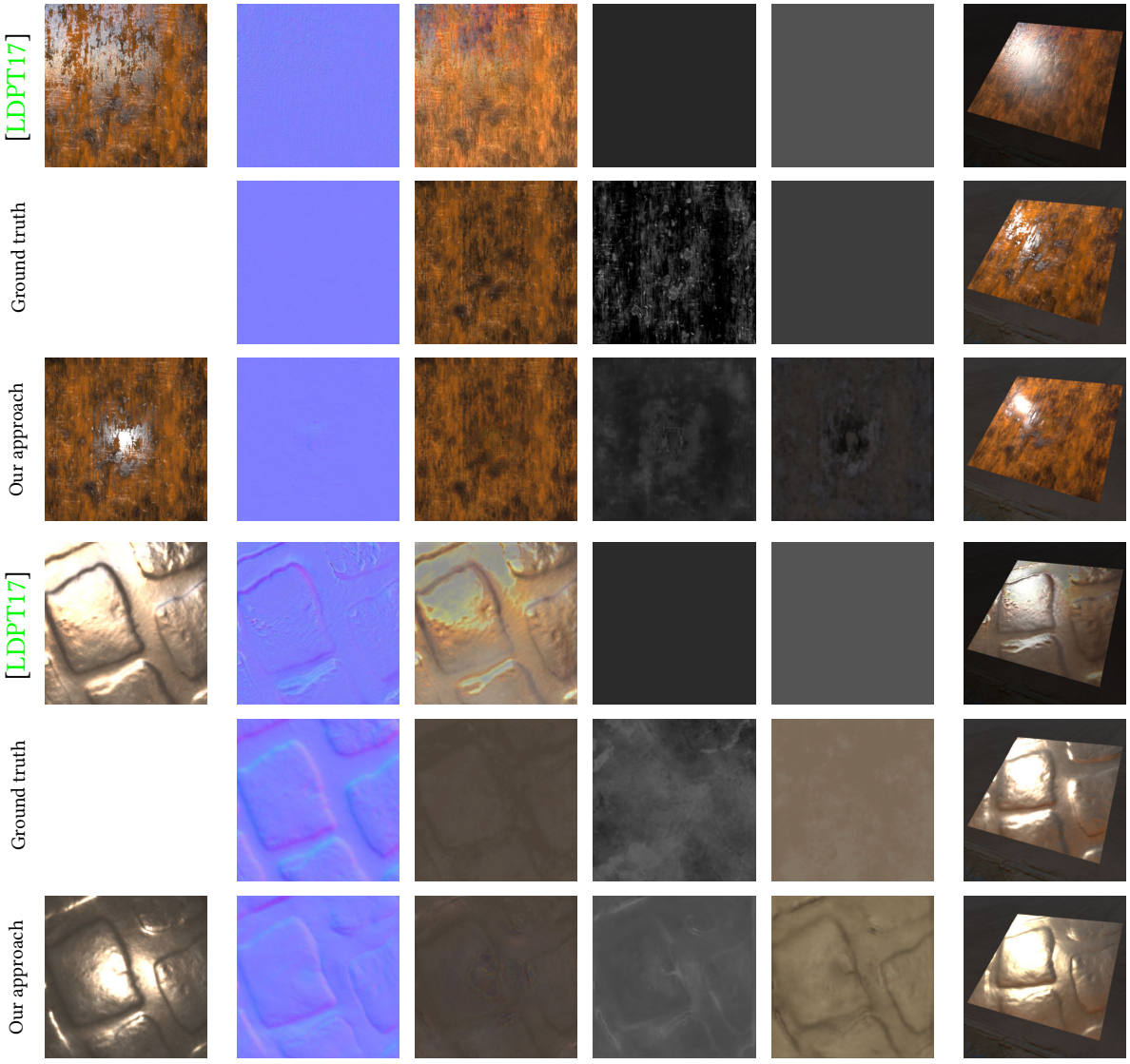


Figure 3.16: Comparison with Li et al. [LDPT17] on synthetic data. As the methods produce output data using different BRDF models, the values of the maps of Li et al. [LDPT17] should not be compared directly to ours or the ground truth. We show them to aid qualitative evaluation of the the spatial variation. To facilitate comparison, we rendered the ground truth and each result under novel illumination conditions (right). The renderings for the results of Li et al. [LDPT17] were made with a lower exposure due to different albedo magnitudes predicted by the methods. The input images (left) were rendered under flash lighting for our method and under environment lighting for the method by Li et al., in agreement with the type of input assumed by each method.

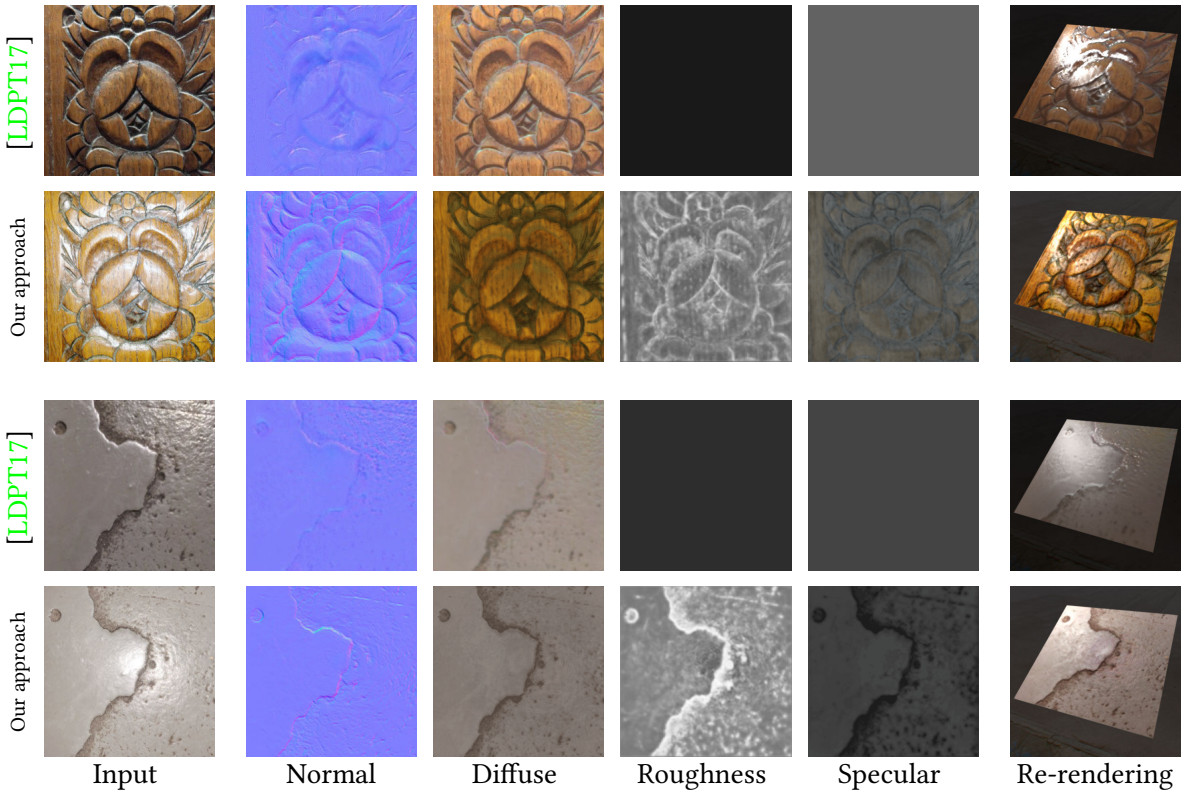
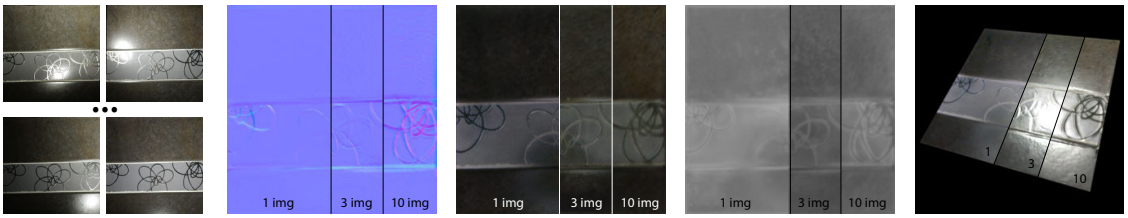


Figure 3.17: Comparison with Li et al. [LDPT17] on real-world data. We captured the input photographs under flash lighting for our method and under environment lighting for the method by Li et al., in agreement with the type of input assumed by each method. Please refer to Figure 3.16 for notes on interpreting the results.

Chapter 4

Flexible SVBRDF Capture with a Multi-Image Deep Network

The work presented in this chapter was done in collaboration with Miika Aittala, Fredo Durand, George Drettakis and Adrien Bousseau and published in the Eurographics Symposium on Renderings 2019 [DAD+ 19].



(a) Input images (b) Normals (c) Diffuse albedo (d) Roughness (e) Re-rendering

Figure 4.1: Our deep learning method for SVBRDF capture supports a variable number of input photographs taken with uncontrolled and uncalibrated light-view directions (a, rectified). While a single image is enough to obtain a first plausible estimate of the SVBRDF maps, more images provide new cues to our method, improving its prediction. In this example, adding images reveals fine normal variations (b), removes highlight residuals in the diffuse albedo (c), and reveals the difference of roughness between the stone, the stripe, and the thin pattern (d). See suppl. materials for animated renderings.

While the method presented in Chapter 3 is able to produce convincing spatially-varying material appearances using deep learning, a single image is often simply not enough to observe the rich appearance of real-world materials. Figure 4.1(b-d) illustrates typical failure cases of single-image methods, where the flash lighting provides insufficient cues of the relief of the surface, and leaves highlight residuals in the diffuse albedo and specular maps. Only additional pictures with side views or lights reveal fine geometry and reflectance details.

We present a deep-learning method capable of estimating material appearance from a variable number of uncalibrated and unordered pictures captured with a handheld camera and flash. The key observation is that such image sets are fundamentally unstruc-

tured. They do not have a meaningful ordering, nor a pre-determined type of content for any given input. Following this reasoning, we adopt a pooling-based network architecture that treats the inputs in a perfectly order-invariant manner, giving it powerful means to extract and combine subtle joint appearance cues scattered across the inputs. This architecture extracts the most useful information from each picture, while benefiting from strong priors learned from data.

We show how our method improves its prediction with the number of input pictures, and reaches high quality reconstructions with as little as 1 to 10 images – a sweet spot between existing single-image and complex multi-image approaches.

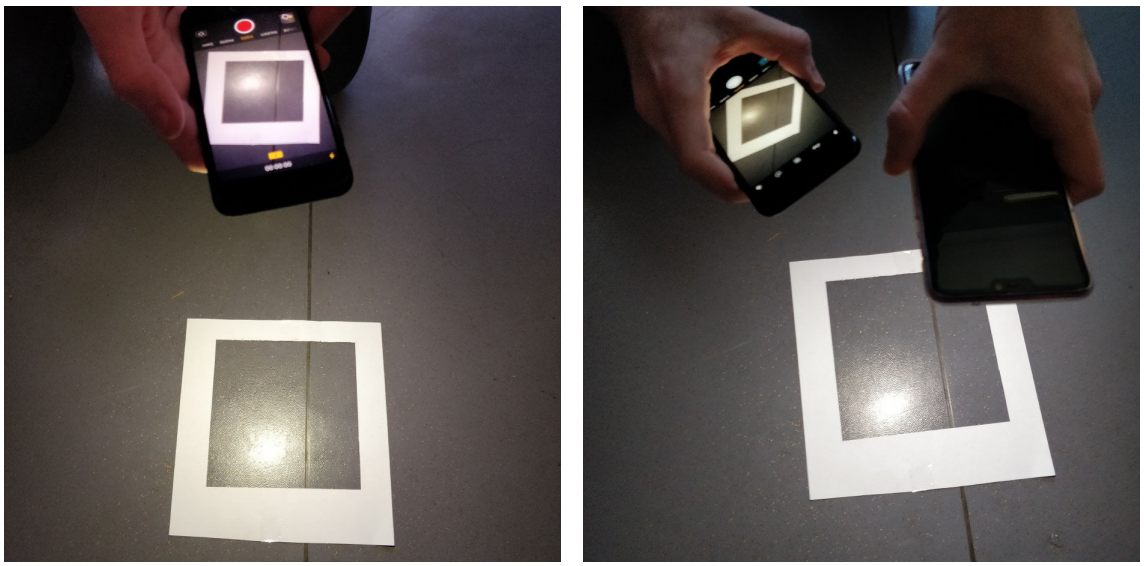


Figure 4.2: We use a simple paper frame to help register pictures taken from different viewpoints. We use either a single smartphone and its flash, or two smartphones to cover a larger set of view/light configurations.

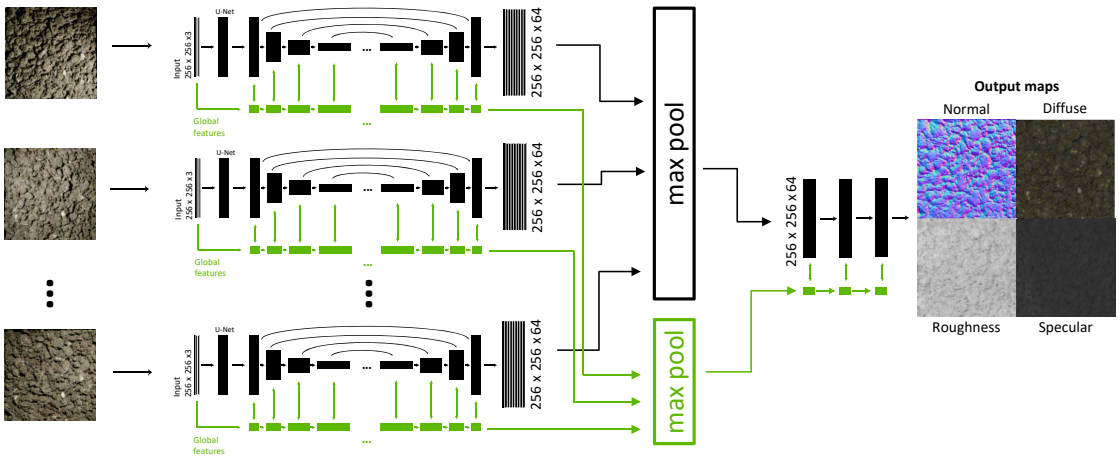


Figure 4.3: Overview of our deep network architecture. Each input image is processed by its copy of the encoder-decoder to produce a feature map. While the number of images and network copies can vary, a pooling layer fuses the output maps to obtain a fixed-size representation of the material, which is then processed by a few convolutional layers to produce the SVBRDF maps.

4.1 Capture Setup

We designed our method to take as input a variable number of images, captured under uncontrolled light and view directions. Figure 4.2 shows the capture setup we experimented with, where we place the material sample within a white paper frame and capture it by holding a smartphone in one hand and a flash in the other, or by using the flash of the smartphone as a co-located light source. Similarly to Paterson et al. [PCF05] and Hui et al. [HSL+17], we use the four corners of the frame to compute an homography that rectifies the images, and crop the paper pixels away before processing the images with our method. We capture pictures of 3456×3456 pixels and resize them to 256×256 pixels after cropping.

4.2 Multi-Image Material Inference

Our goal is to estimate the spatially-varying bi-directional reflectance distribution function (SVBRDF) of a flat material sample given a few aligned pictures of that sample. We adopt a parametric representation of the SVBRDF in the form of four maps representing the per-pixel surface normal and diffuse albedo, specular albedo and specular roughness of a Cook-Torrance [CT82] BRDF model.

The core of our method is a multi-image network composed of several copies of a single-image network [1], as illustrated in Figure 4.3. The number of copies is dynamically chosen to match the number of inputs provided by the user (or the training sample). All copies are identical in their architecture and weights, meaning that each input receives an identical treatment by its respective network copy. The findings from each single-image network are then fused by a common order-agnostic pooling layer before being subsequently processed into a joint estimate of the SVBRDF.

We now detail the single-image network and the fusion mechanism, before describing the loss we use to compare the network prediction against a ground-truth SVBRDF. We detail our generation of synthetic training data in Section 4.3.

4.2.1 Single-image network

We base our architecture on the single-image network of Deschaintre et al. [DAD+18], which was designed for a similar material acquisition task. The network follows the popular U-Net encoder-decoder architecture [RPB15], to which it adds a fully-connected track responsible for processing and transmitting global information across distant pixels. While the original architecture outputs four SVBRDF maps, we modify its last layer to instead output a 64-channel feature map, which retains more information to be processed by the later stages of our architecture. We also provide pixel coordinates as extra channels to the input to help the convolutional network reason about spatial information [LLM+18, LSC18].

Since we are targeting a lightweight capture scenario, we do not provide the network with any explicit knowledge of the light and view position. We rather count on the network to deduce related information from visual cues.

4.2.2 Multi-image fusion

The second part of our architecture fuses the multiple feature maps produced by the single-image networks to form a single feature map of fixed size.

Specifically, the encoder-decoder track of each single-image network produces a $256 \times 256 \times 64$ intermediate feature map corresponding to the input image it processed. These maps are fused into a single joint feature map of the same size by picking the maximum

[1] Source code of our network architecture along with pre-trained weights will be released.

value reported by any single-image network at each pixel and feature channel. This max-pooling procedure gives every single-image network equal means to contribute to the content of the joint feature map in a perfectly order-independent manner [AD18, CHW18]. We explored the use of mean pooling to aggregate information from all images for each pixel, but did not notice significant improvement.

The pooled intermediate feature map is finally decoded by 3 layers of convolutions and non-linearities, which provide the network sufficient expressivity to transform the extracted information into four SVBRDF maps. The global features in the fully-connected tracks are max-pooled and decoded in a similar manner. Through end-to-end training, the single-image networks learn to produce features which are meaningful with respect to the pooling operation and useful for reconstructing the final estimate.

While we vary the number of copies of the single-view network between 1 and 5 during training, an important property of this architecture is that it can process an arbitrarily large number of images during testing because all copies share the same weights, and are ultimately fused by the pooling layer to form a fixed-size feature map. In our experiments, we vary the number of input images from 1 to 10 at testing time.

4.2.3 Loss

We evaluate the quality of the network prediction with a differentiable *rendering loss* [LSC18, LXR+18, DAD+18]. We adopt the loss of Deschaintre et al. [DAD+18], which renders the predicted SVBRDF under multiple light and view directions, and compare these renderings with renderings of the ground-truth SVBRDF under the same conditions. The comparison is performed using an l_1 norm on the logarithmic values of the renderings to compress the high dynamic range of specular peaks.

Following Li et al. [LSC18], we complement this rendering loss with four l_1 losses, each measuring the difference between one of the predicted maps and its ground-truth counterpart. We found this direct supervision to stabilize training. Our final loss is a weighted mixture of all losses, $L = L_{\text{Render}} + 0.1\big(L_{\text{Normal}} + L_{\text{Diffuse}} + L_{\text{Specular}} + L_{\text{Roughness}}\big)$.

4.2.4 Training

We train our network for 7 days on a Nvidia GTX 1080 TI. We let the training run for 1 million iterations with a batch size of 2 and input sizes of 256×256 pixels. We use the

Adam optimizer [KB15] with a learning rate set to 0.0002 and $\beta = 0.5$.

4.3 Online Generation of Training Data

Following prior work on deep-learning for inverse rendering [RGR+17, LDPT17, DAD+18, LSC18, LXR+18, LCY+17], we rely on synthetic data to train our network. While in theory image synthesis offers the means to generate an arbitrary large amount of training data, the cost of image rendering, storage and transfer limits the size of the datasets used in practice. For example, Li et al. [LSC18] and Deschaintre et al. [DAD+18] report training datasets of $150{,}000$ and $200{,}000$ images respectively. This practical challenge motivated us to implement an online renderer that generates a new SVBRDF and its multiple renderings at each iteration of the training, yielding up to 2 million training images in practice.

We first explain how we generate numerous ground-truth SVBRDFs, before describing the main features of our SVBRDF renderer.

4.3.1 SVBRDF synthesis

We rely on procedural, artist-designed SVBRDFs to obtain our training data. Starting from a small set of such SVBRDF maps, Deschaintre et al. [DAD+18] perform data augmentation by computing $20{,}000$ convex combinations of random pairs of SVBRDFs. We follow the same strategy, although we implemented this material mixing within TensorFlow [AAB+15], which allows us to generate a unique SVBRDF for each training iteration while only loading a small set of base SVBRDFs at the beginning of the training process. We use the dataset proposed by Deschaintre et al., which contains $1,850$ SVBRDFs covering common material classes such as plastic, metal, wood, leather, *etc*, all obtained from Allegorithmic Substance Share [All19b].

4.3.2 SVBRDF rendering

We implemented our SVBRDF renderer in TensorFlow, so that it can be called at each iteration of the training process. Since our network takes rectified images as input, we do not need to simulate perspective projection of the material sample. Instead, our renderer simply takes as input four SVBRDF maps along with a light and view position, and evaluates the resulting rendering equation at each pixel. We augment this basic renderer

with several features that simulate common effects encountered in real-world captures:

Viewing conditions. We distribute the camera positions over an hemisphere centered on the material sample, and vary its distance by a random amount to allow a casual capture scenario where users may not be able to maintain an exact distance from the target. We also perform random perturbations of the field-of-view (set to $40°$ by default) to simulate different types of cameras. Finally, we apply a random rotation and scaling to the SVBRDF maps before cropping them to 256×256 pixels, which simulates materials of different orientations and scales.

Lighting conditions. We simulate a flash light as a point light with angular fall-off. We again distribute the light positions over an hemisphere at a random distance to simulate a handheld flash. Other random perturbations include the angular fall-off to simulate different types of flash, the light intensity to simulate varying exposure, and the light color to simulate varying white-balance. Finally, we also include the simulation of a surrounding lighting environment in the form of a second light with random position, intensity and color, which is kept fixed for a given input SVBRDF.

Image post-processing. We have implemented several common image degradations – additive Gaussian noise, clipping of radiance values to 1 to simulate low-dynamic range images, gamma correction and quantization over 8 bits per channel.

While rendering our training data on the fly incurs additional computation, we found that this overhead is compensated by the time gained in data loading. In our experiments, training our system with online data generation takes approximately as much time as training it with pre-computed data stored on disk, making the actual rendering virtually free.

4.4 Results and Evaluation

We evaluate our method using a dataset of 32 ground truth SVBRDFs not present in the set used for training data generation. We also use measured Bidirectional Texture Functions (BTFs) [WGK14] to compare the re-renderings of our predictions to real-world appearances. Finally, we used our method to acquire a set of around 80 real materials.

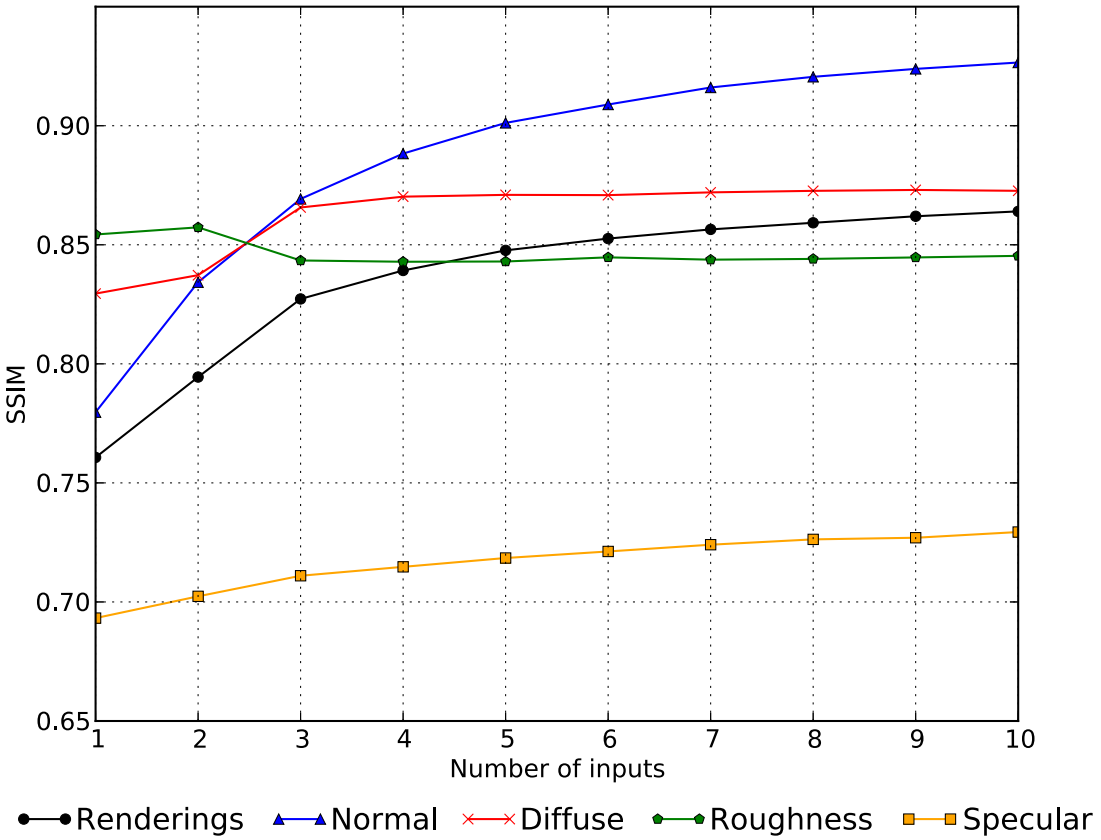


Figure 4.4: SSIM of our predictions with respect to the number of input images, averaged over our synthetic test dataset. The SSIM of re-renderings increases quickly for the first images, before stabilizing at around 10 images. The normal maps strongly benefit from new images. Diffuse and specular albedos also improve with additional inputs, which is not the case of the roughness that remains stable overall. We provide similar RMSE plots as supplemental materials.

Since our method does not assume a controlled lighting, we used either the camera flash or a separate smartphone as the light source for those acquisitions. All results in the figures of this chapter were taken with two phones; please see supplemental for all results and examples acquired with a single phone. Resulting quality is similar in both cases.

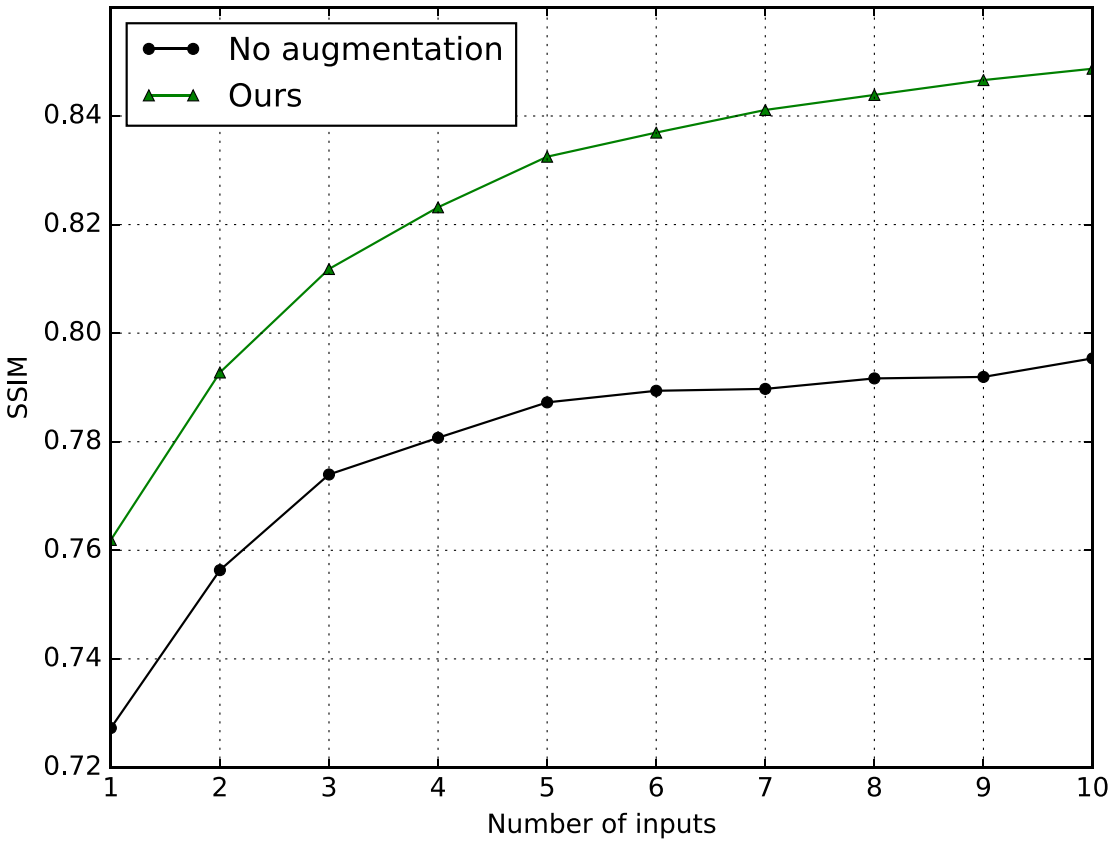


Figure 4.5: Ablation study. Comparison of SSIM between our method (green) and a restricted version (black) where the network is trained with lighting and viewing directions chosen on a perfect hemisphere, and with all lighting parameters constant (falloff exponent, power, etc.). Our complete method achieves higher SSIM when tested on a dataset with small variations of these parameters, showing that it is robust to such perturbations that are frequent in casual real world capture.

4.4.1 Number of input images

A strength of our method is its ability to cope with a variable number of photographs. We first evaluate whether additional images improve the result using synthetic SVBRDFs, for which we have ground truth maps. We measure the error of our prediction by re-rendering our predicted maps under many views and lights, as done by the rendering loss used for training. Figure 4.4 plots the SSIM similarity metric of these re-renderings averaged over the test set for an increasing number of images, along with the SSIM of the individual SVBRDF maps. While most improvements happen with the first five

images, the similarity continues to increase with subsequent inputs, stabilizing at around 10 images. The diffuse albedo is the fastest to stabilize, consistent with the intuition that few measurements suffice to recover low-frequency signals. Surprisingly, the quality of the roughness prediction seems on average independent of the number of images, suggesting that the method struggles to exploit additional information for this quantity. In contrast, the normal prediction improves with each additional input, as also observed in our experiments with real-world data detailed next. We provide RMSE plots of the same experiment as supplemental materials.

Using the same procedure, in Figure 4.5 we perform an ablation study to evaluate the impact of including random perturbations of the viewing and lighting conditions in the training data. As expected, the network trained without perturbation does not perform as well as our complete method on our test dataset that includes view and light variations similar to those in casual real world capture. We trained both networks for $750{,}000$ iterations for this experiment.

Figure 4.6 shows our predictions on a measured BTF material from the Bonn database [WGK14], using 1, 2, 3 and 10 inputs. For this material, normals, diffuse albedo and roughness estimations improve with more inputs. In particular, the normal map progressively captures more relief, the diffuse albedo map becomes almost uniform, and the embossed part on the upper right is quickly recognized as shinier than the remaining of the sample.

For a real material capture we performed (Figure 4.7), we see similar effects: normals are improved with more inputs, and the difference of roughness between different parts is progressively recovered. However, we do not have access to ground truth maps for these real-world captures.

Overall, our results in Figure 4.4-4.10 and in supplemental material illustrate that our method achieves our goals: adding more pictures greatly improves the results, notably removing artifacts in the diffuse albedo while improving normal estimation. Our method enhances the quality of recovered materials while maintaining a casual capture.

4.4.2 Comparison to multi-image optimization

We compare our data-driven approach to a traditional optimization that takes as input multiple images captured under the assumption of known and precisely calibrated light

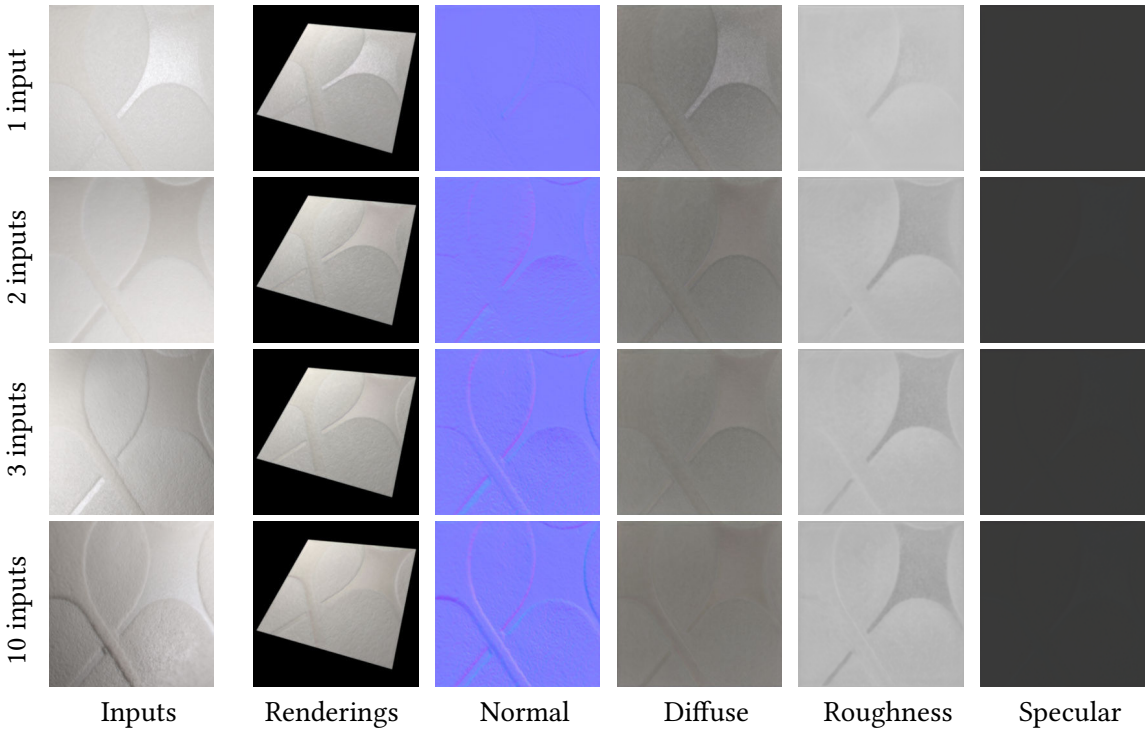


Figure 4.6: Evaluation on a measured BTF. Three images are enough to capture most of normal and roughness maps. Adding images further improves the result by removing lighting residual from the diffuse albedo, and adding subtle details to the normal and specular maps.

and viewing conditions. Given these conditions we solve for the SVBRDF maps that minimize re-rendering error of the input images, as measured by our rendering loss. We further regularize this optimization by augmenting the loss with a total-variation term that favors piecewise-smooth maps. We solve the optimization with the Adam algorithm [KB15]. While the optimization stabilizes after 900K iterations, we let it run for a total of 2M iterations to ensure full convergence, which takes approximately 3.5 hours on an NVIDIA GTX 1080 TI. Given the non-convex nature of the optimization, we initialize the solution to a plausible estimate obtained by setting the diffuse albedo map to the most fronto-parallel input, the normal map to a constant vector pointing upward, the roughness to zero and the specular albedo to gray. We use synthetic data for this experiment, which provides us with full control and knowledge of the viewing and lighting conditions needed by the optimization, as well as with ground truth maps to evaluate the quality of the outcome.

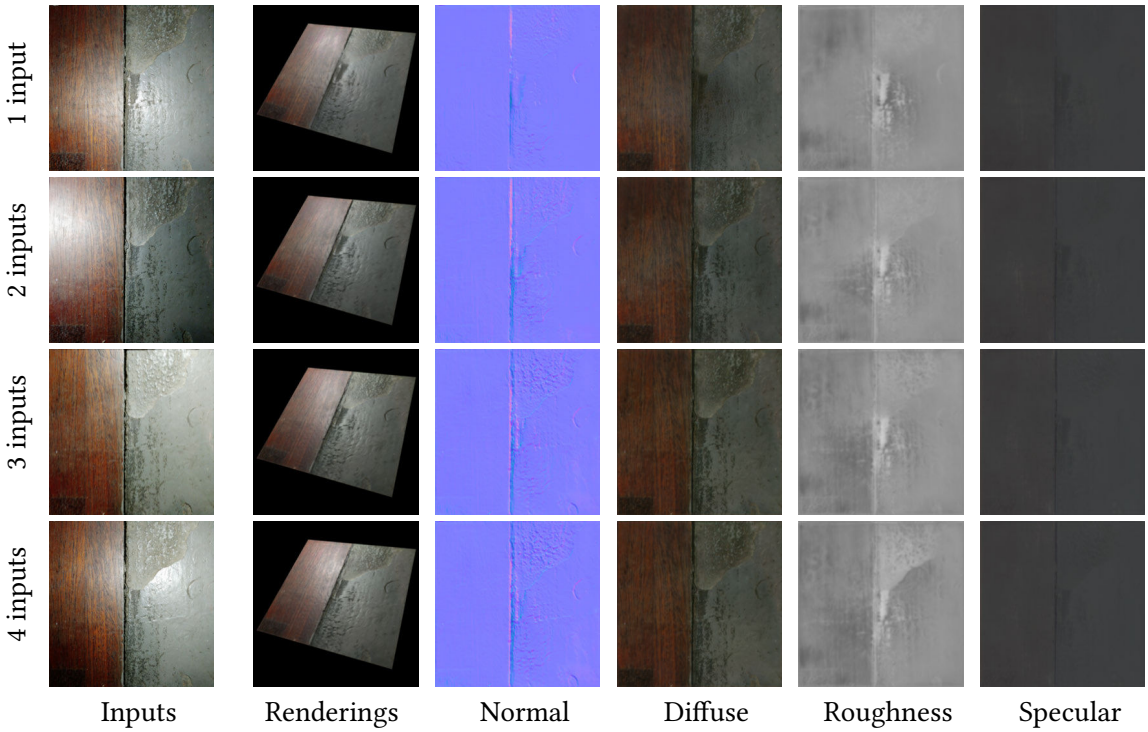


Figure 4.7: A single flash picture hardly provides enough information for surfaces composed of several materials. In this example, adding images allows the recovery of normal details, and the capture of different roughness values in different parts of the image. Note in particular how the 4th image helps capturing a discontinuity of the roughness on the right part.

Figure 4.8 compares the number of input images required to achieve similar quality between the classical optimization and our method, using view and light directions uniformly distributed over the hemisphere. On rather diffuse materials (stones, tiles), the optimization needs a few dozen calibrated images to achieve a result of similar quality to the one produced by our method using only 5, uncalibrated images. A similar number of images is necessary for a material with uniform shininess (scales). However more than 900 images were necessary for our optimization to reach the quality obtained by our method on a material with significant normal and roughness variations (wood). Overall, our method achieves plausible results with much fewer inputs captured under unknown lighting, although classical optimization can recover more precise SVBRDFs if provided with enough carefully-calibrated images.

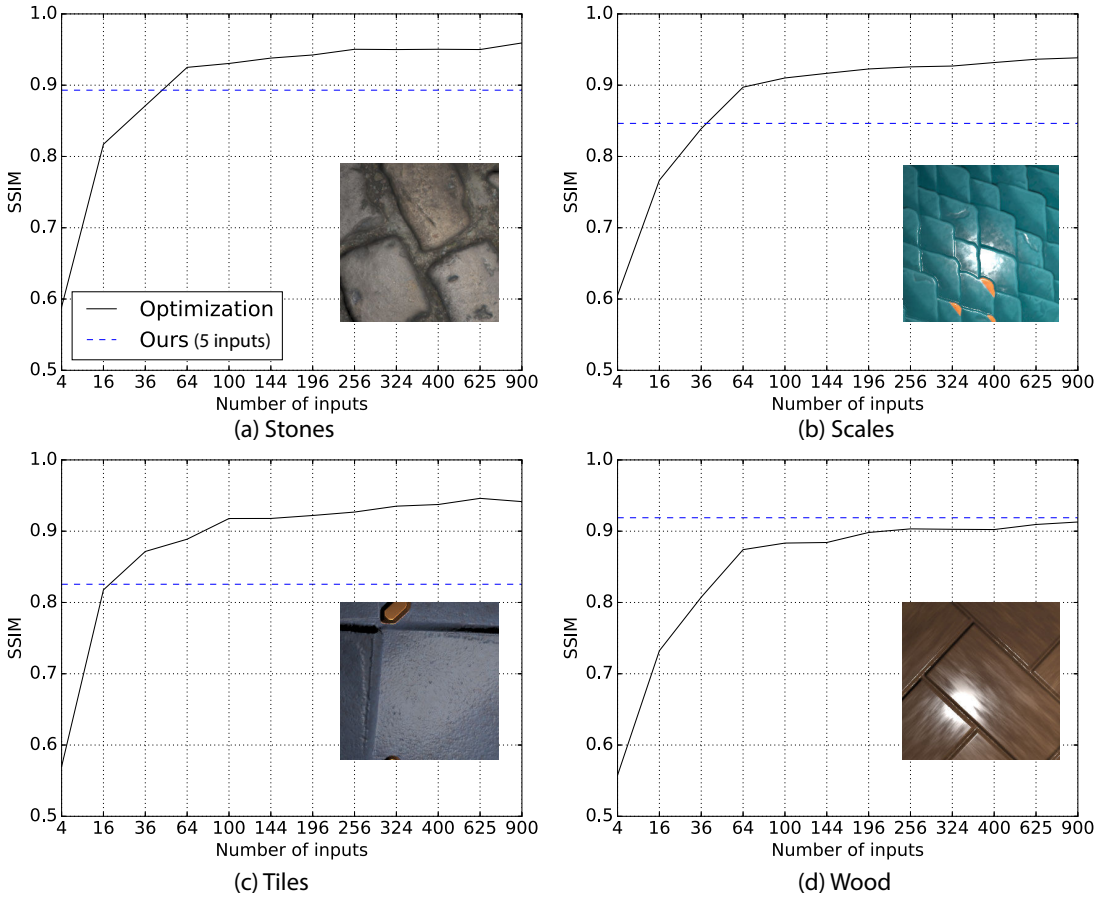


Figure 4.8: SSIM on re-renderings for the maps obtained by our method with 5 images (dotted blue) and by a classical optimization method with an increasing number of input images (black). The classical optimization requires several dozens of calibrated pictures to outperform our method on rather diffuse or uniform materials (stones, tiles, scales), while requiring many more for a more complex material (wood).

4.4.3 Comparison to alternative deep learning methods

We first compare our architecture to a simple baseline composed of the network by Deschaintre et al. [DAD+18] augmented to take 5 images instead of one. This baseline achieves an average SSIM of 0.826, similar to the SSIM of 0.847 produced by our method for the same number of inputs. This evaluation demonstrates that our multi-image network performs as well as a fixed network while providing the freedom to vary the number of input images.

We next compare to the recent single-image methods of Deschaintre et al. [DAD+18] and Li et al. [LSC18], which both take as input a fronto-parallel flash photo. Figure 4.9 provides a visual comparison on synthetic SVBRDFs with ground truth maps, Figure 4.11 provides a similar comparison on BTFs measured from 81x81 pictures, which allow ground-truth re-renderings, and Figure 4.10 provides a comparison on real pictures. While developed concurrently, both single-image approaches suffer from the same limitations. The co-located lighting tends to produce low-contrast shading, reducing the cues available for the network to fully retrieve normals. Adding side-lit pictures of the material helps our approach retrieve these missing details. The fronto-parallel flash also often produces a saturated highlight in the middle of the image, which both single-image methods struggle to in-paint convincingly in the different maps. While the strength of the highlight could be reduced by careful tuning of exposure, saturated pixels are difficult to avoid in real-world capture. In contrast, our method benefits from additional pictures to recover information about those pixels.

Another limitation of these two single-image methods is that the flash highlight cannot cover all parts of the material sample. This lack of information can cause erroneous estimations, especially when the sample is composed of multiple materials with different shininess. Providing more pictures gives a chance to our method to observe highlights over all parts of the sample, as is the case in Figure 4.7, where the difference in roughness in the upper right only becomes apparent with the 4th input.

4.4.4 Limitations

Since our method builds on the single-image network of Deschaintre et al. [DAD+18], it inherits some of its limitations. First, the method is limited to materials that can be well represented by an isotropic Cook-Torrance BRDF. We also observe that the method tends to produce correlated maps and interpret dark materials as shiny, as shown in Figure 4.12(top) where despite several pictures, albedo variations of the cardboard get interpreted as normal variations, and the black letters get assigned a low roughness. This behavior reflects the content of our training data, since most artist-designed SVBRDFs have correlated maps.

Since we rectify the multi-view inputs with a simple homography, we do not correct for parallax effects produced by surfaces with high relief. This approximation may yield misalignment in the input images, which in turn reduces the sharpness of the predicted

maps. In addition, our SVBRDF representation, training data, and rendering loss do not model cast shadows. While shadows are mostly absent in pictures taken with a co-located flash, they can appear when using a handheld flash and remain visible in some of our results, as shown in Figure 4.12 (bottom).

Finally, our experiments on synthetic data suggest that providing additional images improves quality on all SVBRDF parameters except roughness (Figure 4.4). This limitation may be partly due to the use of a point light source, which produces small specular highlights, and to the uncertainty in the light distance and properties. Differentiable rendering of other light sources (e.g., area lights or environment maps) might address this partly, making capture somewhat more flexible and potentially improving performance since gloss is better conveyed by extended lights [FDA03].

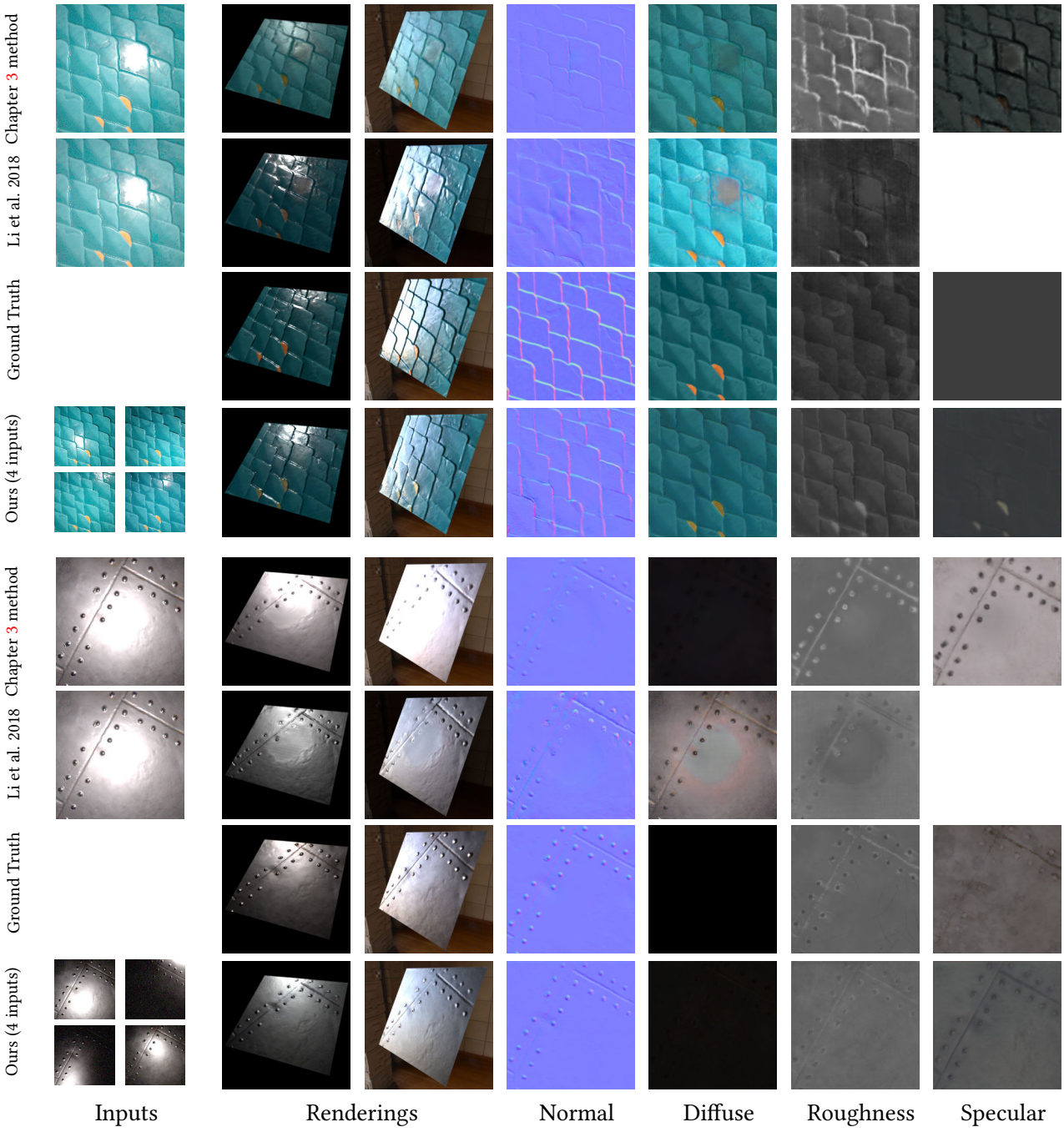


Figure 4.9: Comparison against single-image methods on synthetic SVBRDFs. Our method leverages additional input images to obtain SVBRDF maps closer to ground truth. In particular, single-image methods under-estimate normal variations and fail to remove the saturated highlight on shiny materials. See supplemental materials for more comparisons and results.

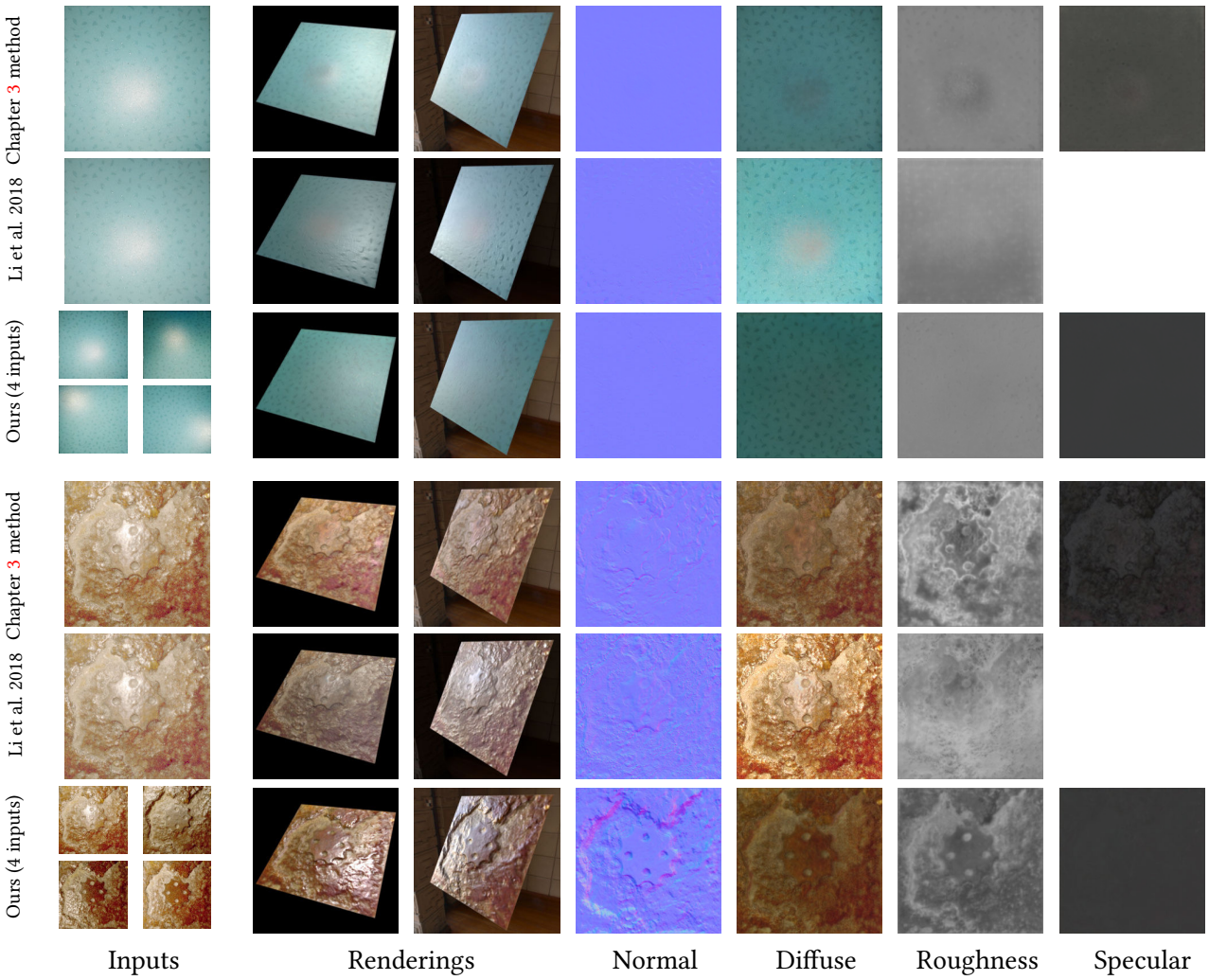


Figure 4.10: Comparison against single-image methods on real-world pictures. Our method recovers more normal details, and better removes highlight and shading residuals from the diffuse albedo. See supplemental materials for more comparisons and results.

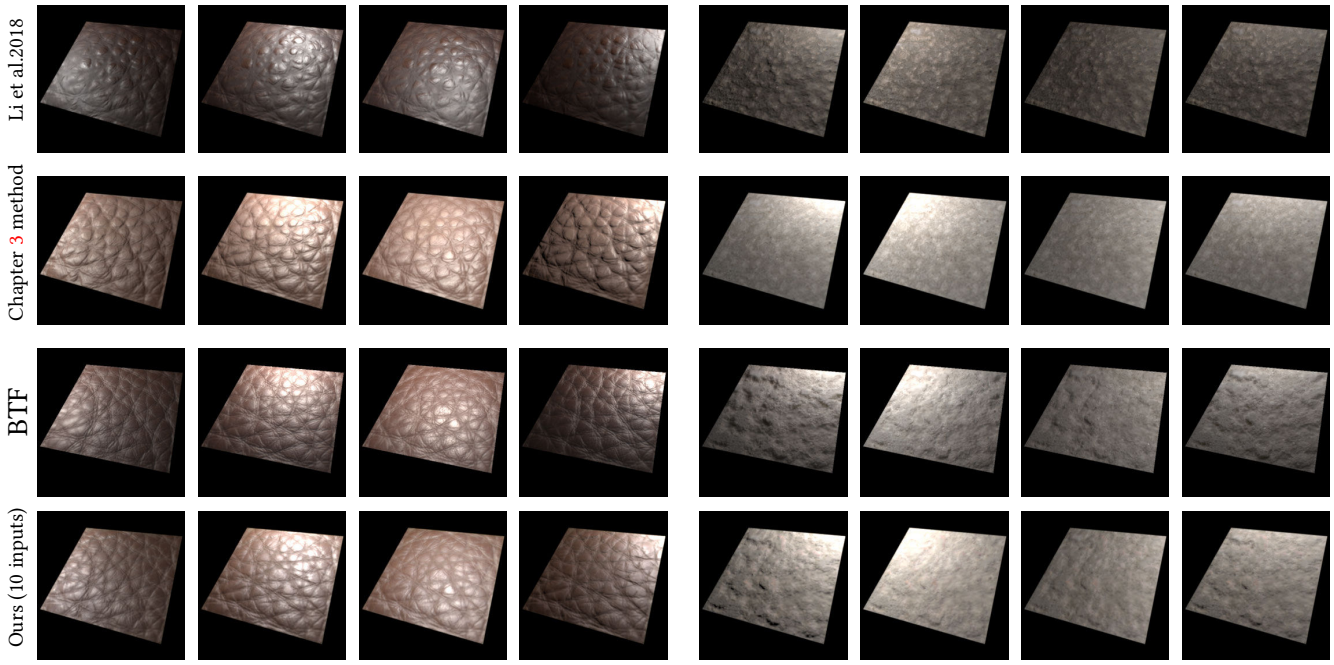


Figure 4.11: Comparison against single-image methods on a measured BTF with ground truth re-renderings. Our method globally captures the material features better.

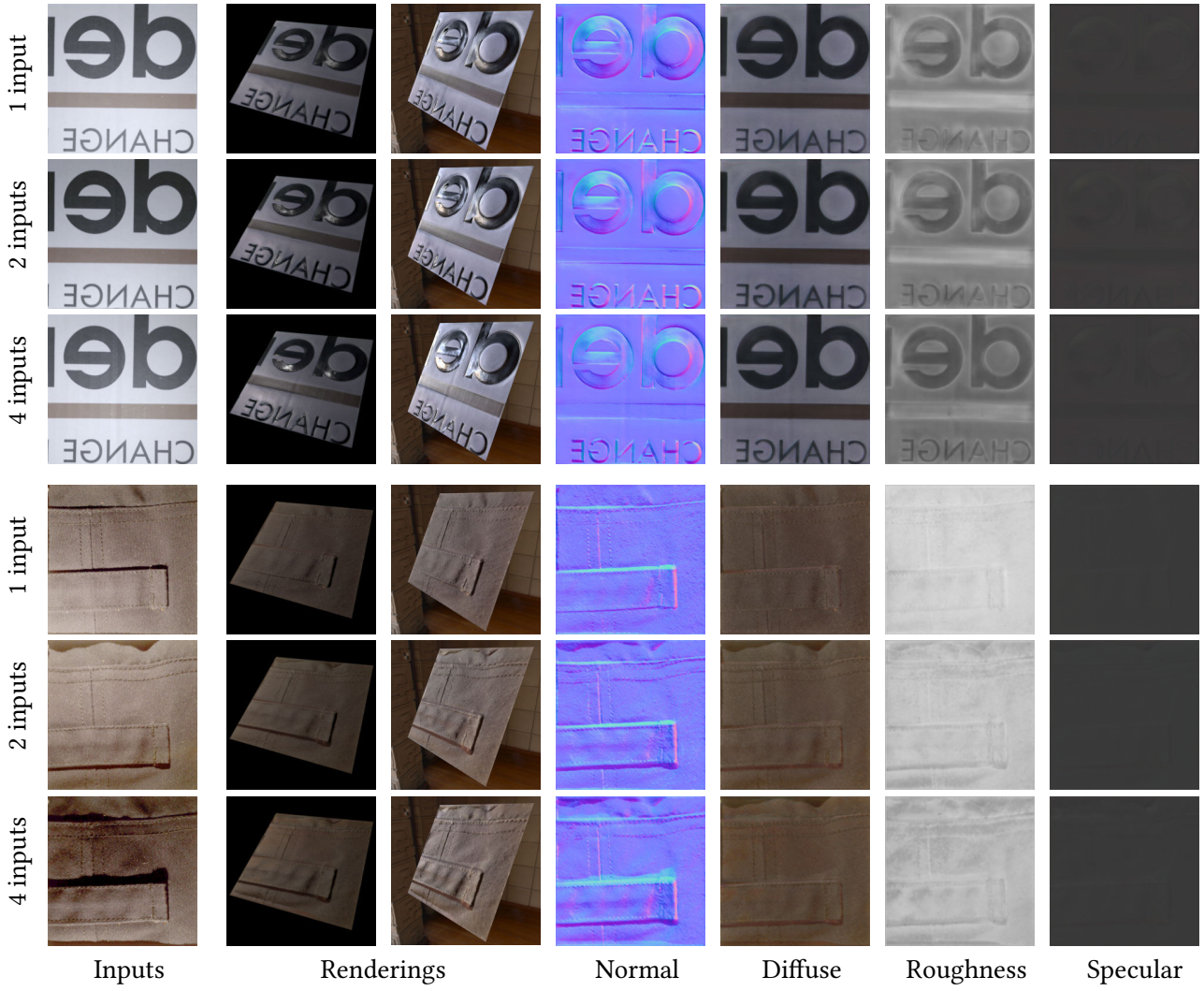


Figure 4.12: Limitations. We inherits some of the limitations of the method by Deschaintre et al. [DAD+18], such as the tendency to produce correlated maps and to interpret dark pixels as shiny (top). Our SVBRDF representation, training data and loss do not model cast shadows. As a result, shadows in the input pollute some of the maps (bottom).

4.5 Conclusion

In this chapter, we described a data generation pipeline allowing for fast experimentation on data simulating different acquisition process. Using our tensorflow based renderer, we are able to transfer the time cost of reading data on a cluster to rendering new material at each training step.

We also present a deep network architecture allowing to combine an arbitrary number of pictures to retrieve high quality SVRBDF parameters. We are nonetheless still limited to low resolution for GPU memory issues and small patches of materials as the acquisition picture must be taken from close enough to maintain the flash power. In Chapter 5 we explore how to solve this limitation, and increase both the resolution and the scale of the acquired surface.

Chapter 5

By-Example Capture of Large-Scale SVBRDFs

The work presented in this chapter was done in collaboration with Adrien Bousseau and George Drettakis and is under submission.

Figure 5.1: Our method estimates the SVBRDF of large surfaces from just a few pictures. In a typical capture session, users only need to take a single picture of the entire surface, along with a small number of close-ups. This lightweight workflow is particularly advantageous for on-site capture, as was the case for the shiny mural and tiled floor shown above. Alternatively, users can feed our method with an existing picture, along with a small number of exemplar SVBRDF patches of similar materials. This second workflow provides a new form of artistic control to material designers. Please see supplemental materials for high-resolution SVBRDF parameter maps and animated renderings of all our results, which give a much better impression of the material properties.

Recent progress on lightweight appearance capture allows the recovery of real-world spatially-varying reflectance (SVBRDF) from just a few photographs of a surface. In particular, multiple methods —such as the ones described in Chapters 3 and 4 and recent work [AWL15, AAL16, RPG16, HSL+17, DAD+18, LSC18, DAD+19, GLD+19] — take as input one or several photographs captured with a hand-held camera, where the co-located flash provides informative spatially-varying illumination over the measured surface sample.

In this chapter, we complement such small-scale inputs with a picture of the entire surface, taken under ambient lighting. Our method then fuses these two sources of information to propagate the SVBRDFs estimated from each close-up flash picture to all pixels of the large image. To achieve this goal, we designed a deep neural network that takes as input the large-scale image and an arbitrary number of small-scale exemplar SVBRDF patches, and outputs a large SVBRDF aligned with the input image. This design leverages the complementary strengths of deep-learned priors and inference-time exemplars. Furthermore, we describe how to decompose the large input into independent tiles, which allows our method to process images that cannot fit entirely on GPU memory. Thanks to our two-scale approach, we can capture surfaces several meters wide, such as walls, doors and furniture.

We demonstrate the strength of our approach in two usage scenarios. In our first scenario – *on-site acquisition* – we capture a single photograph of a large surface as well as a few close-up flash photographs of its details. We then use our method, described in Chapter 3 to extract SVBRDF maps from the flash photographs [DAD+18], and use our method to propagate this information to the large image, effectively acquiring SVBRDFs several meters wide. In our second scenario – *creative design* – we use stock photographs as our large-scale input, and artist-created SVBRDFs already available to us as our exemplars, demonstrating fine control on the creation of realistic SVBRDFs solely from existing data. This new workflow offers users the ability to control the materials assigned to the large image by selecting different exemplars, greatly enhancing the creative possibilities offered by capture-based materials.

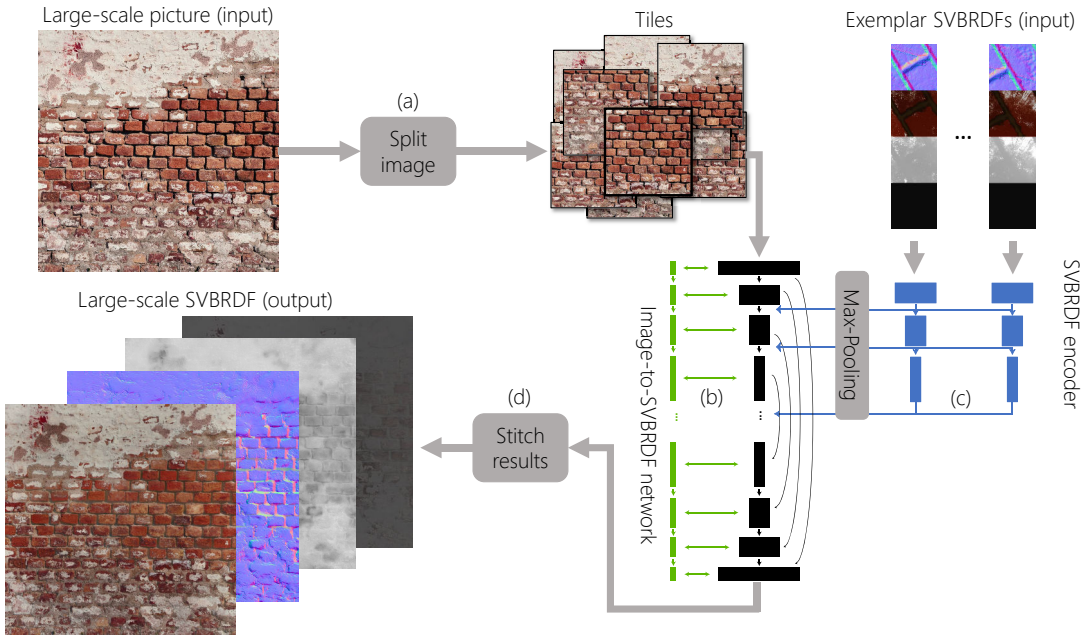


Figure 5.2: Overview of our method. We first split the input large-scale image into overlapping tiles (a). Each tile is then processed by a U-Net encoder-decoder to produce SVBRDF maps (b, black blocks). This convolutional network communicates with a fully-connected network that extracts and processes global features of the material, as proposed in Chapter 3 (b, green blocks). We complement this architecture with one or several SVBRDF encoders (c) that process the material exemplars. Features extracted by these encoders are fused and injected at multiple levels of the U-Net encoder, and in turn transmitted to the U-Net decoder via its skip connections (b, black arrows). The last step of our method is to stitch the SVBRDF maps predicted from all tiles to form a large-scale output (d).

5.1 Method

Our method takes as input a single picture of a large-scale surface captured under ambient lighting, along with a few close-up SVBRDF patches of the same or a similar surface. These patches can be acquired with any existing small-scale SVBRDF capture method, or taken from a library of real-world or synthetic SVBRDFs, possibly including different variants of the material. Intuitively, the large scale image represents the target surface for which we want to estimate the SVBRDF, while the close-up patches serve as exemplars of what this SVBRDF should look like locally. Our method supports a varying number of exemplar patches, for instance to treat large-scale surfaces composed of a

mixture of small-scale materials. The close-up patches also offer users a way to control the SVBRDF reconstruction in ambiguous cases where multiple valid interpretations exist. For example, in the absence of highlights, the picture of a brick wall could equally be interpreted as made of diffuse bricks or of specular ceramic tiles. We demonstrate that we can achieve either of these results by providing diffuse or specular exemplar patches to our method.

The main task of our algorithm is to propagate information from a few exemplar patches to all pixels of the large-scale image. Related work on guided image synthesis tackle a similar challenge by building explicit correspondences between the target image and the exemplars [MGSJW12] or between reflectance intensity of a limited number of materials to transfer roughness properties [RPG16]. However, our experiments reveal that this strategy tends to fail when the exemplars are only representative of parts of the target image, as is common in our application scenario. We instead propose a novel approach based on deep learning, which allows our method to combine the information present in the exemplar patches with material priors learned from a large dataset of SVBRDFs.

Our deep network is composed on two main branches. The first branch processes the large-scale image in an encoder-decoder fashion, similar to our network described in Chapter 3 and recent work on single-image SVBRDF capture using deep learning [DAD+18, LSC18]. The other branch encodes each SVBRDF patch into a compact descriptor. We aggregate the information extracted from all patches using max pooling layers, which allows our method to handle an arbitrary number of exemplar patches in an order-independent manner. Finally, we concatenate the feature maps computed by the SVBRDF encoder to these computed by the large-scale image encoder, allowing the subsequent decoder to exploit the two sources of information. Figure 5.2 provides a visual overview of our method to extract SVBRDF parameter maps for large-scale surfaces. We first describe typical inputs to our method, before explaining our deep neural network architecture and training process.

5.1.1 Inputs

Our goal is to generate SVBRDF parameter maps for large-scale planar surfaces, such as walls, doors or furniture. To do so, our method takes two forms of input. First, a single picture of the surface of interest, captured under ambient indoor or outdoor lighting. Second, a series of SVBRDF patches that represent small parts of the surface, or of a

similar material. To obtain these patches, we either capture close-up flash pictures of the surface and run our single-image SVBRDF method described in Chapter 3, or we select SVBRDFs from a library of artist-designed materials [All19b].

As a pre-process, we split the large-scale image into tiles of 512×512 pixels to fit in GPU memory. Neighboring tiles have an overlap of 256 pixels to facilitate subsequent stitching of their SVBRDF maps. We assume that all tiles receive approximately the same lighting, which is not the case for pictures taken with a flash as used in our methods described in Chapters 3 and 4 and prior work [DAD+18, LSC18, DAD+19, GLD+19].

Finally, we also render each exemplar SVBRDF patch under a random distant lighting and provide these images as extra channels to help the method relate the input large-scale image with the exemplars.

5.1.2 Neural network architecture and loss

Our method processes each tile of the input image independently to output four Cook-Torrance [CT82] SVBRDF maps, corresponding to the normal, diffuse albedo, specular albedo, and specular roughness of each input pixel. This task is performed by an encoder-decoder convolutional neural network similar to the one used in Chapter 3 and recent single-image methods [DAD+18, LSC18]. In particular, we adopt the architecture described in Chapter 3, where a convolutional U-Net [RPB15] computes image features at multiple scales while a fully-connected network extracts and transmits global information across scales.

However, a single image often does not provide enough information to recover SVBRDF parameters unambiguously, especially in the absence of flash highlights. Our solution to this challenge is to complement the image encoder-decoder with an SVBRDF encoder, which extracts multi-scale features from an exemplar SVBRDF patch. We then inject this additional information into the image network by concatenating the feature maps extracted by the SVBRDF encoder with the feature maps of same resolution extracted by the image encoder. The features concatenated at each level are processed by the next level of the U-Net encoder, and are also transmitted to the corresponding level of the decoder thanks to the U-Net skip connections. Importantly, we train the image and SVBRDF deep networks jointly, such that the SVBRDF encoder learns to extract multi-scale features that best help the image encoder-decoder in its SVBRDF estimation task.

An additional difficulty raised by our target application is that different images might require a different number of exemplar patches to cover their constituent materials. We designed our method to support an arbitrary number of exemplars by aggregating the feature maps extracted by several instances of the SVBRDF encoder into a single pyramid of feature maps. We perform this aggregation using max pooling over the set of feature maps, which selects the strongest activations independently of the order in which the patches are processed. A similar mechanism has been recently used for related problems, such as multi-image material acquisition, described in Chapter 4 [DAD+19], photometric stereo [CHW18], and burst image deblurring [AD18].

Similarly to our work described in Chapter 4 and recent work on deep material capture [DAD+19, LSC18], we train our network to minimize a *reconstruction loss* that measures the per-pixel L1 distance between each predicted parameter map and its ground truth, as well as a *rendering loss* that measures the per-pixel L1 distance between renderings of the predicted SVBRDF and ground truth under 9 different view and light conditions. In addition, we also account for the local structure of the material by evaluating a multi-scale Structural SIMilarity metric (SSIM) on these renderings.

5.1.3 Post-processing

The last step of our method consists in merging the predictions of all tiles into a large-scale SVBRDF. Since all tiles are processed using the same exemplars, neighboring tiles mostly agree in their predictions up to low frequency variations. We achieve a seamless composite by blending the tiles over their overlap using a Gaussian weighting kernel that gives a weight of 1 at the center of the tile and reaches almost 0 at its border. This mechanism allows our method to be applied on high-resolution inputs of arbitrary aspect ratio.

5.1.4 Training

We trained our deep network with synthetic SVBRDFs [All19b] that provide ground truth supervision for each parameter map. We rely on the same set of training SVBRDFs as described in Chapter 3, except that we render them at a higher resolution of 2048×2048 pixels. At each training step, a unique material is created by mixing two of these pre-computed SVBRDFs. We then extract between 1 and 3 patches of 256×256 pixels to form the SVBRDF exemplars. Finally, we extract a large crop to serve as the input image,

which we resize to 512×512 pixels to be fed to the network. We vary the size of this crop between 512×512 and 1024×1024 pixels, such that the input image and SVBRDF exemplars contain features of different scales. Similarly to Chapter 4, we perform all these processing steps at training time in TensorFlow [AAB+15] to reduce storage and data transfer.

We trained the network for 700 000 steps to obtain the results shown in this paper and supplemental materials, which took around 6 days on a GV100 graphics card.

5.2 Evaluation

We first present results obtained by applying our method on our own photographs as well as on internet images. We then compare our method with alternative approaches on synthetic data for which we have ground truth SVBRDF maps. Please see supplemental materials for high-resolution SVBRDF parameter maps and animated renderings of all our results, which give a much better impression of the material properties.

5.2.1 Results

Our research was originally motivated by the need to quickly acquire the appearance of large-scale surfaces with minimal hardware. Following this first usage scenario, we used a smartphone to photograph a variety of planar objects (walls, floors, furniture). For each object, we first captured a single photograph showing the object in its entirety, under

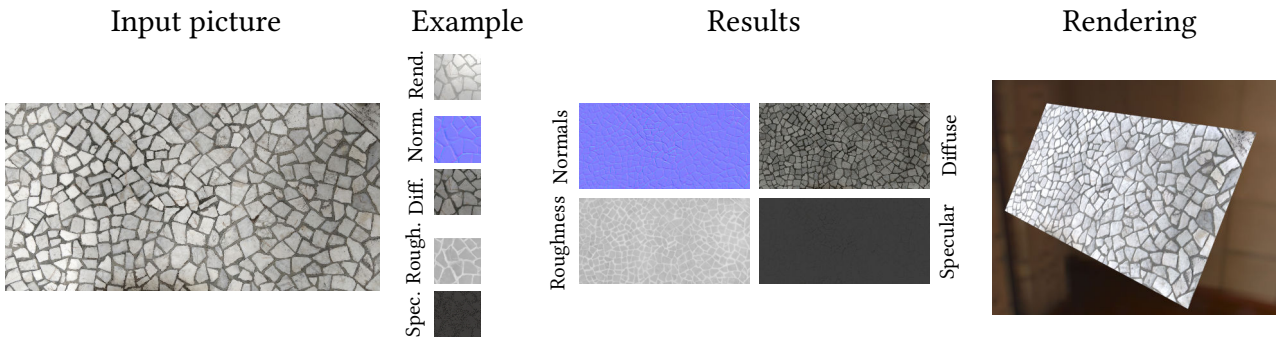


Figure 5.3: Real-world surface captured on-site with our method. We used a single flash picture to capture the shininess of the tiles, which is propagated to all tiles of the large floor. Please zoom on the .pdf to appreciate the high-resolution details of the individual SVBRDF maps.

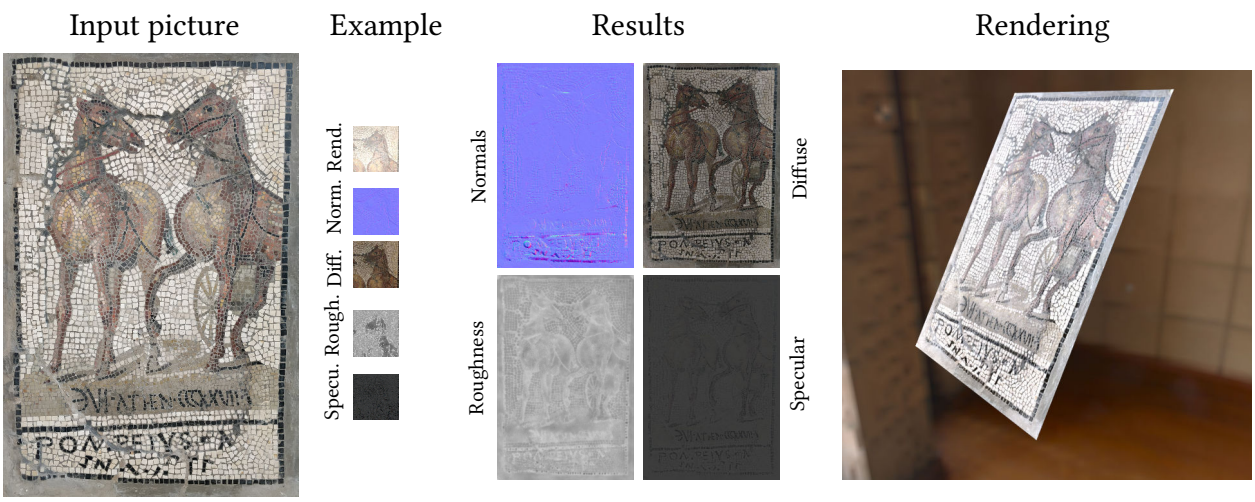


Figure 5.4: Real-world surface captured on-site with our method. We used a single flash picture to capture the shininess of the tiles, which is propagated to all tiles of the mosaic. Please zoom on the .pdf to appreciate the high-resolution details of the individual SVBRDF maps.

ambient lighting. We then captured one to three close-up flash photographs of parts that exhibit characteristic material features. Finally, we ran the single-image SVBRDF estimation network described in Chapter 3 to obtain SVBRDF exemplars for each close-up. Figure 5.1, 5.3 and 5.4 show tiled floors, an ornamental mural, and a mosaic captured on-site with this approach. Thanks to the provided exemplars, our method faithfully reproduces the varying shininess of the different tiles, as well as the metallic appearance of the golden mural.

A second usage scenario of our method is to estimate the SVBRDF maps of existing pictures, using existing SVBRDFs as exemplars of similar materials. Figure 5.5 illustrates this workflow on three internet images, which we processed with exemplars taken from results of our method described in Chapter 3 or from a library of artist-created procedural SVBRDFs [All19b]. Our method transfers the relief and shininess of the exemplars across the surface while conforming to the input image. Note for instance how the roughness map predicted for the brick wall (Figure 5.5, 3rd row) makes the red bricks shinier than the paint, yet includes variations correlated with the presence of dust.

Figure 5.6 further demonstrates the influence of the input exemplars on the output SVBRDF. In this example, the SVBRDF patches obtained from flash close-ups are quite rough. Replacing these exemplars by a synthetic SVBRDF of shiny tiles lowers the rough-

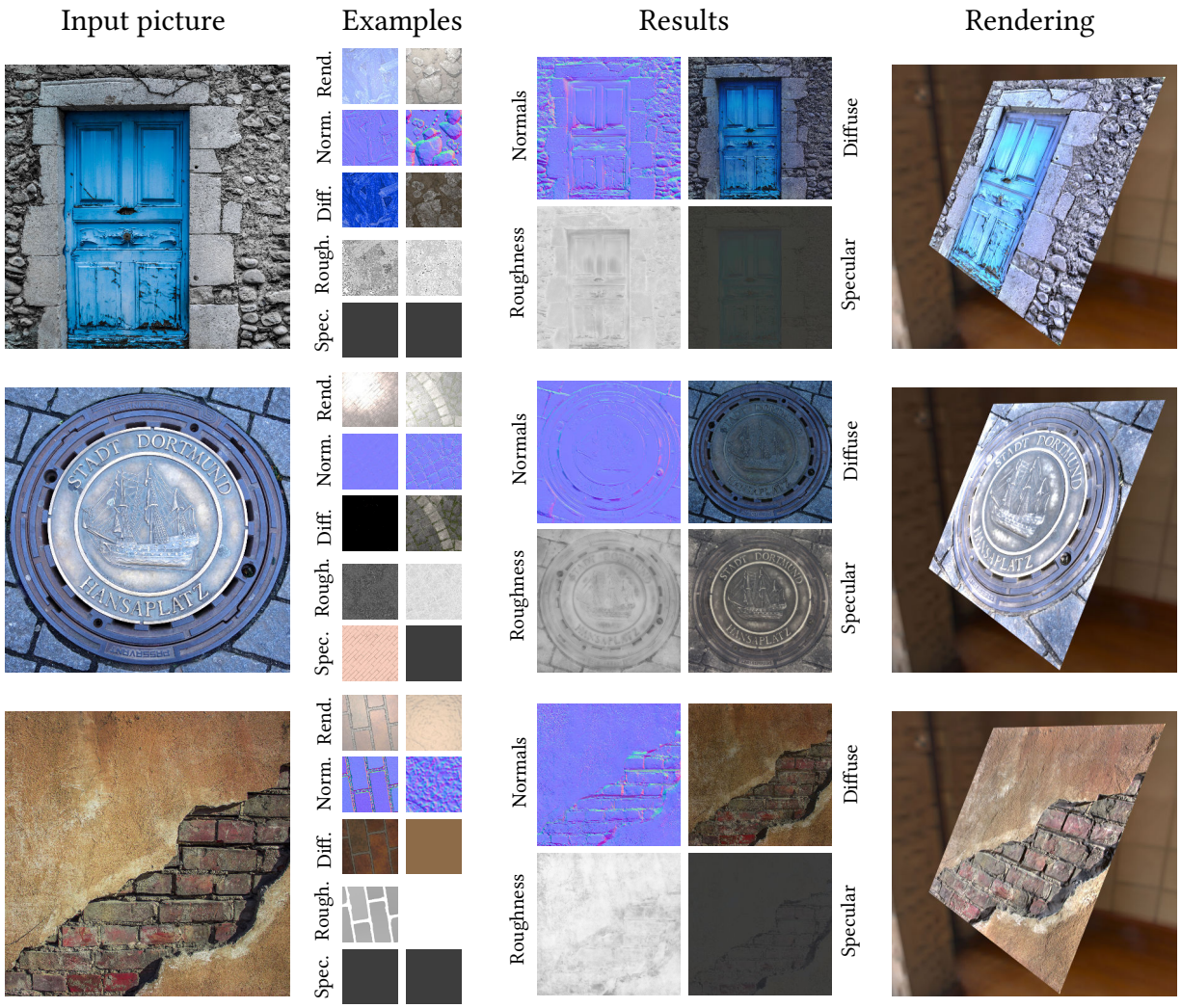


Figure 5.5: Various SVBRDFs estimated from internet images. We selected captured or procedural SVBRDF patches as exemplar materials, which helps our method recover the spatially-varying normals and roughness of stones, bricks, metal, paint. Please zoom on the .pdf to appreciate the high-resolution details of the individual SVBRDF maps.

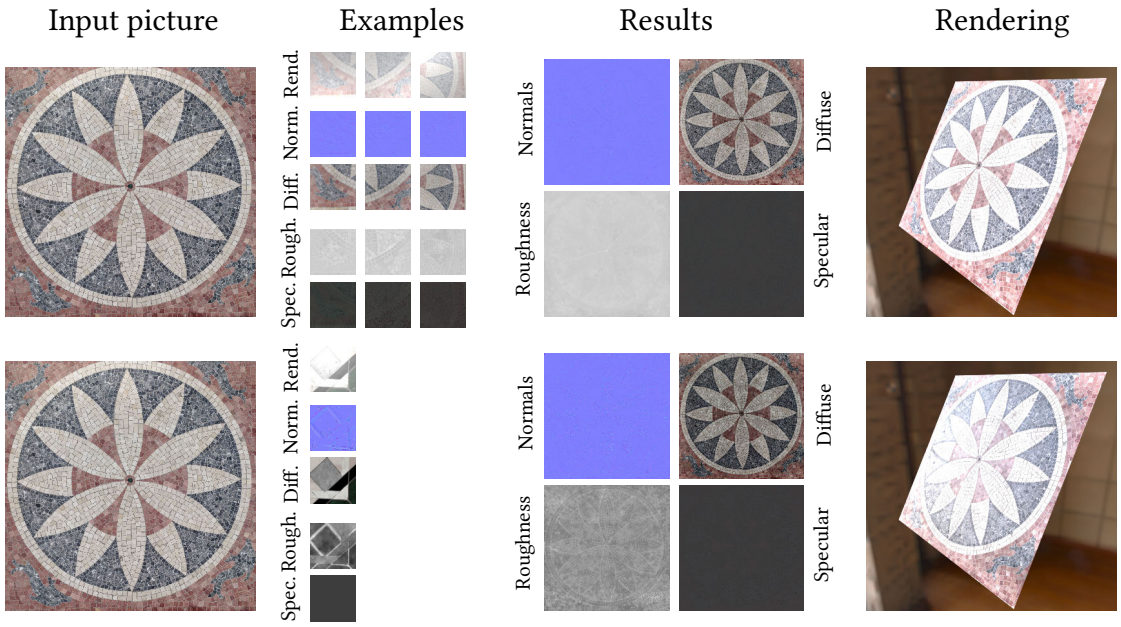


Figure 5.6: Given the same input picture, we achieve different outcomes by changing the exemplars. The exemplars in top row are less shiny than in the bottom row, which effectively translates to the predicted roughness map, and to the final rendering.

ness of the mosaic, yielding sharper highlights in the rendering.

Finally, Figure 5.10 showcases a variety of SVBRDFs created with our method, either via on-site acquisition or from stock photographs. Note that most of these results represent large, non-square surfaces encoded as high-resolution parameter maps, which contrasts with the small material samples often shown in related work.

5.2.2 Comparisons

To our knowledge, our method is the first to offer by-example guidance to deep SVBRDF inference. We first compare to a baseline without exemplars, before comparing to related work on style transfer. We use synthetic SVBRDFs for these comparisons.

Our method uses the single-image network described in Chapter 3 as a backbone for SVBRDF prediction. Figure 5.7 shows results of their method when trained on our dataset. In the absence of flash highlights, the single-image network alone under-estimates the material roughness, and tends to interpret variations of the diffuse albedo as normal variations. In contrast, our method transfers the overall appearance of the exemplar, yielding a stronger roughness and a flatter normal map, in accordance with the ground

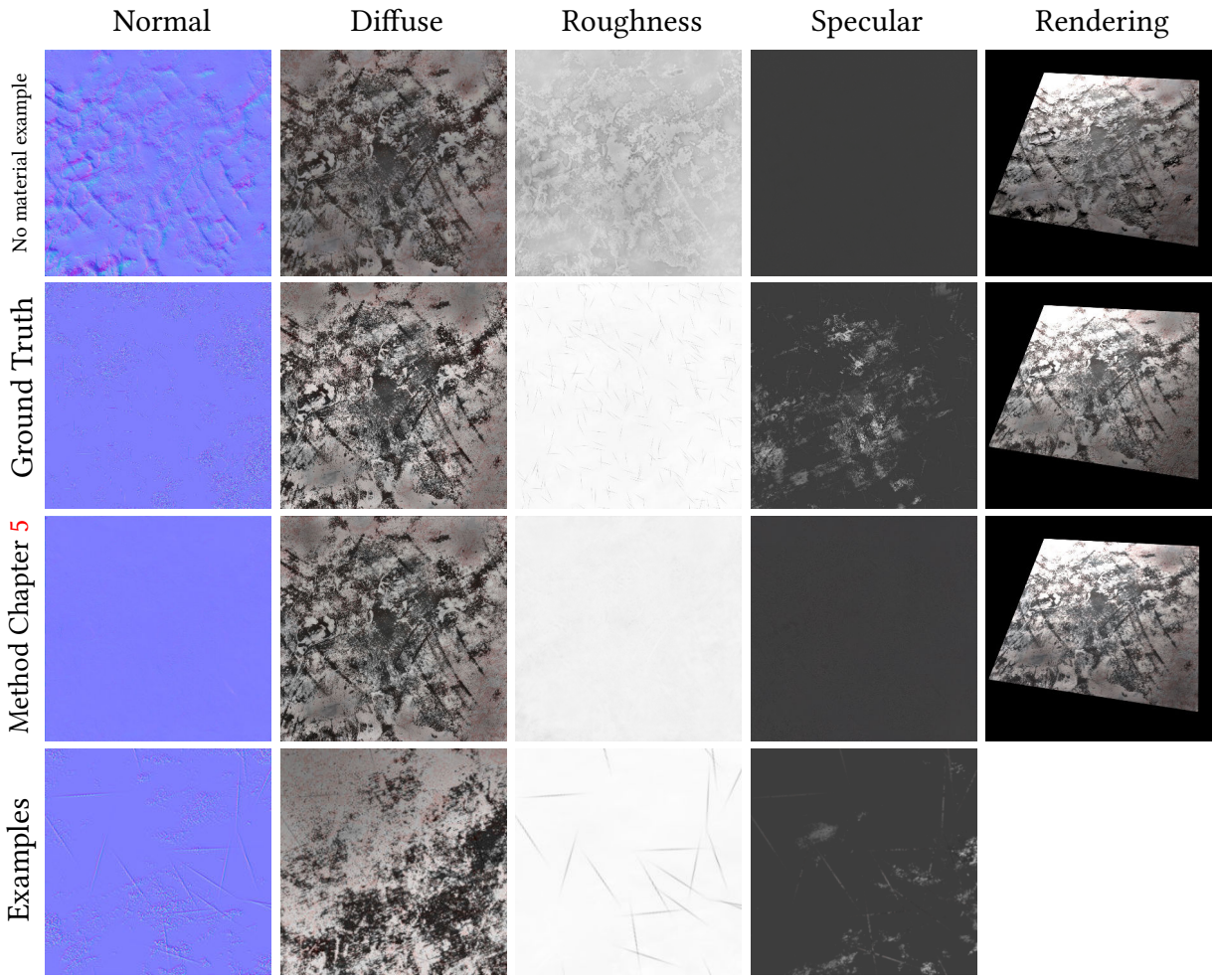


Figure 5.7: Comparison with our single-image method presented in Chapter 3 (top) trained on images under ambient lighting. Our example-based approach better reproduces the flat normals and high roughness of this material.

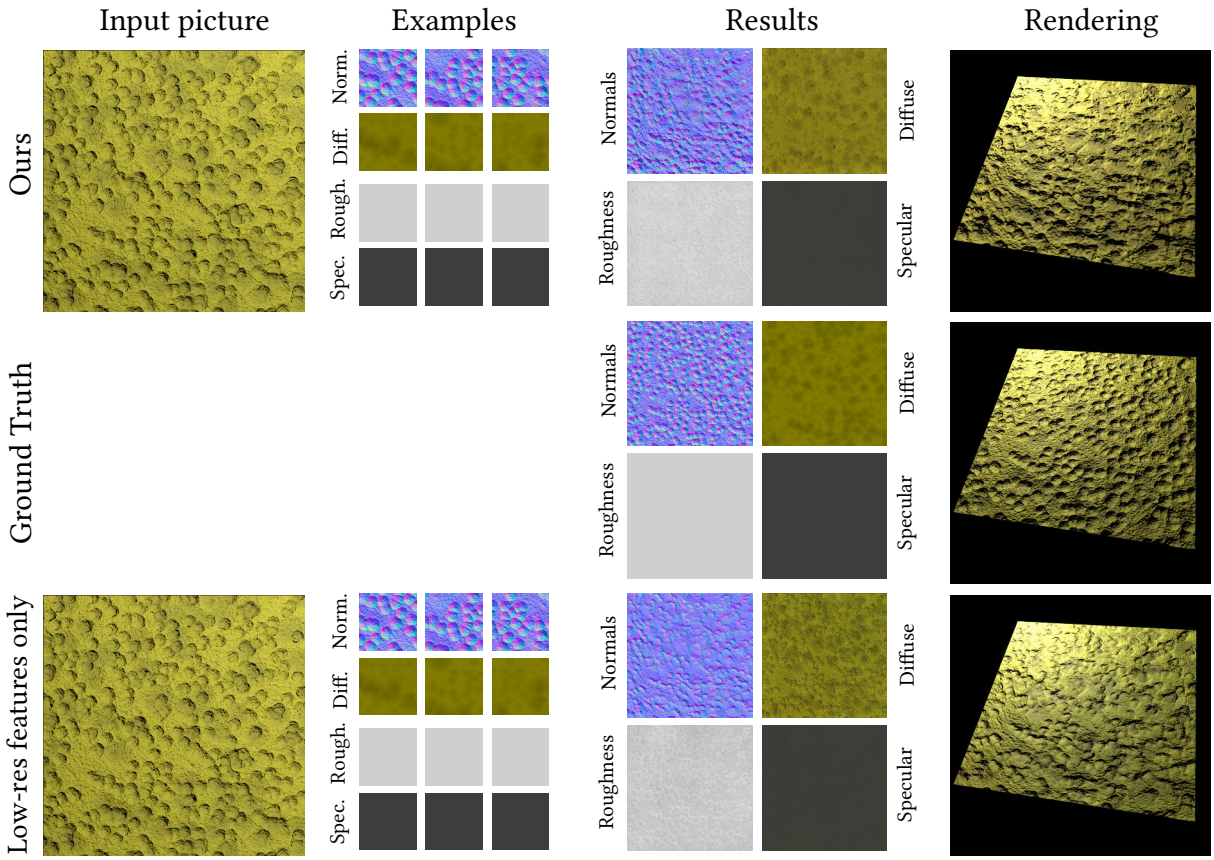


Figure 5.8: Comparison to a variant of our method that only transfers the low-resolution feature maps of the SVBRDF exemplar. Our multi-scale design better captures local variations of normals, roughness and diffuse albedo.

truth. However, both the baseline and our method struggle to capture the small, fine scratches, which are hardly visible even in ground-truth re-renderings.

Figure 5.8 provides a comparison against a variant of our method, where we only transfer the lowest-resolution feature map of the SVBRDF encoder. This ablation reveals the benefits of transferring multi-scale features, in particular for the recovery of fine details.

Our approach is most related to the method by Melendez et al. [MGSJW12], which transfers diffuse albedo and displacement maps using a patch-based texture synthesis algorithm akin to image analogies [HJO+01]. We reproduced this approach using the state-of-the-art patch-based synthesis algorithm of Fišer et al. [FJL+16], using the rendered SVBRDF as guidance. Note that since this algorithm was originally developed for style

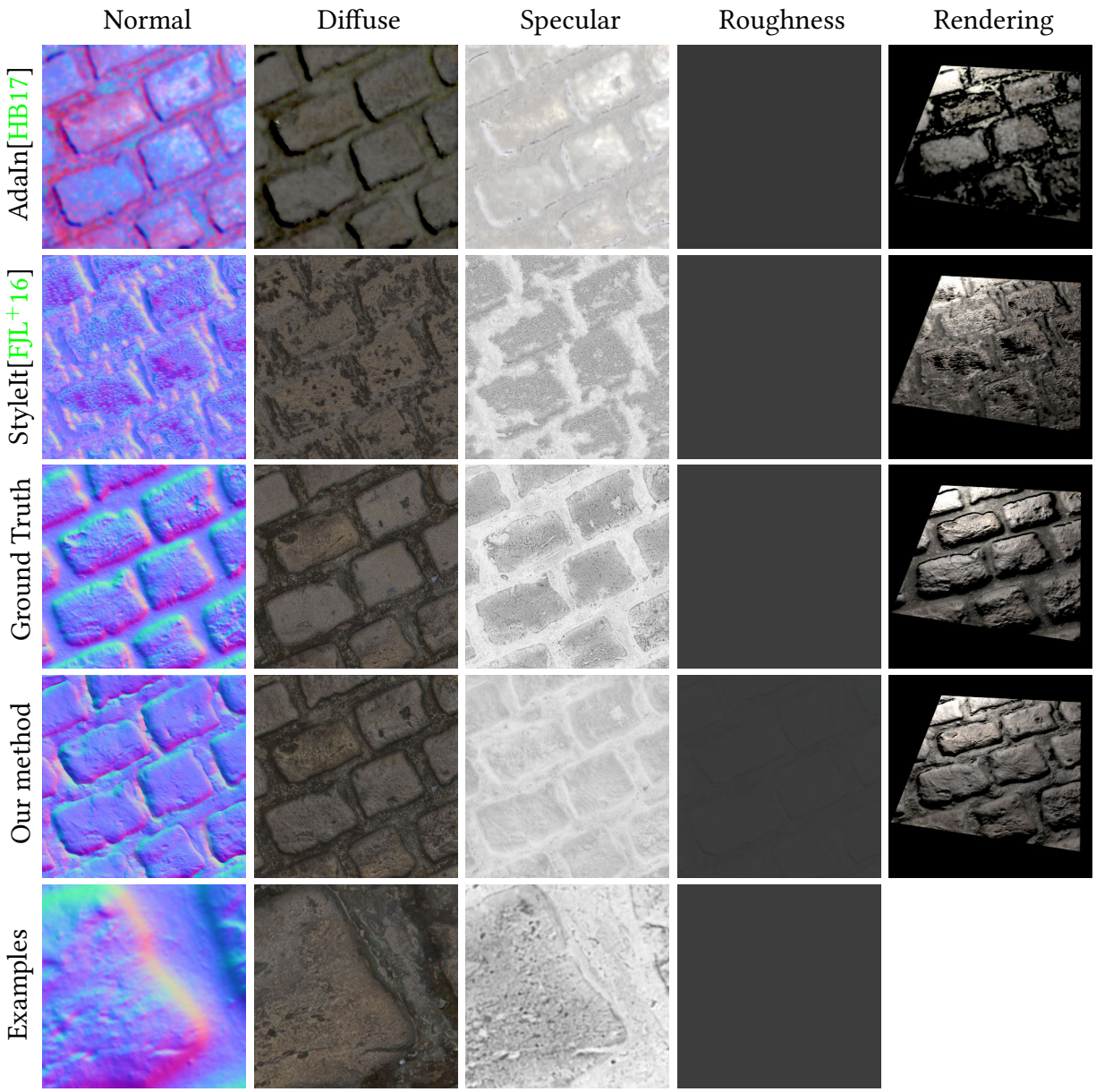


Figure 5.9: Comparison to neural style transfer [HB17] and patch-based texture synthesis [FJL+16]. Our method better transfers details of the surface compared to prior work, which either only captures global statistics (1st row) or struggles to generalize from a limited exemplar (2nd row).

transfer, it assumes that the image to be synthesized only contains three color channels. We coped with this limitation by running their code on each SVBRDF map separately. Figure 5.9 provides the results of this experiment, where the patch-based synthesis lacks variety in the maps due to the limited variety of the provided exemplar. While more advanced synthesis algorithms have been proposed to interpolate between limited exemplars [DBP+15], our method based on deep learning natively generalizes the exemplar to the entire large-scale image.

Finally, Figure 5.9 also includes a comparison to AdaIN [HB17], a recent stylization algorithm based on deep learning that transfers statistics of deep features between an exemplar image and a target. Similarly to the above experiment, we applied the pre-trained method on each SVBRDF parameter map separately. While this generic style transfer algorithm reproduces the overall color distribution of the maps, it misses many of the fine details.

5.2.3 Limitations

As with previous deep-learning based methods for material capture, including the ones described in Chapters 3 and 4 [DAD+18, DAD+19, LSC18], we cannot handle cast shadows, or any other phenomenon that requires more than a normal/bump map. Extending our approach to handle such cases, e.g., using a displacement map, would require a much more complex learning pipeline to handle 3D and the consequent complexities of rendering during training.

Our method has difficulty to distinguish different materials that share similar colors, such as the shiny leather and rough wood in Figure 5.11, which are interpreted as having a similar average roughness. A possible solution to this limitation might be to augment the input image and exemplars with semantic guidance channels, as is commonly done for style transfer [FJS+17]. Our method also assumes that the large-scale input is captured under largely uniform lighting When this is not the case, large illumination gradients pollute the SVBRDF maps, as shown in Figure 5.12.

Figure 5.10: A variety of surfaces captured with our method.

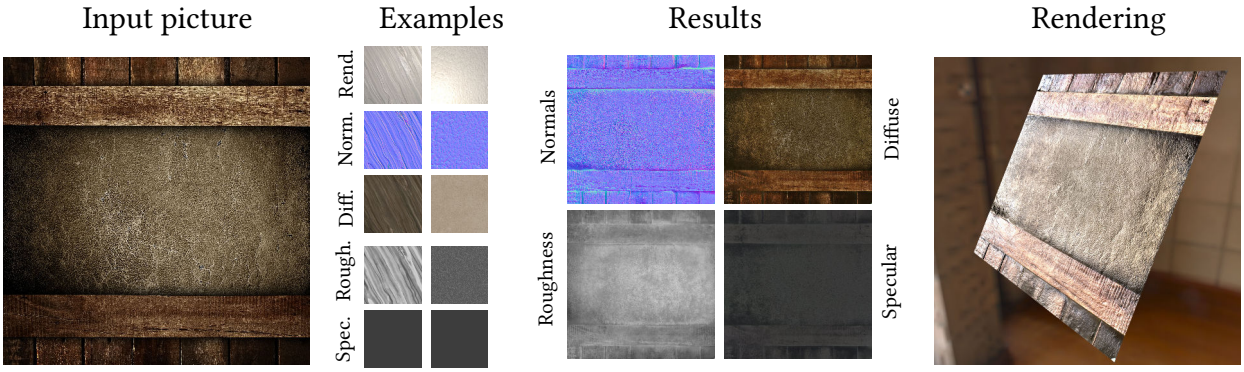


Figure 5.11: Limitation. In the absence of additional guidance, our method can have difficulty distinguishing materials with similar colors, like this wood and leather. While we selected a diffuse wood and a shiny leather as exemplars, the predicted roughness map assigns similar values to the two materials.

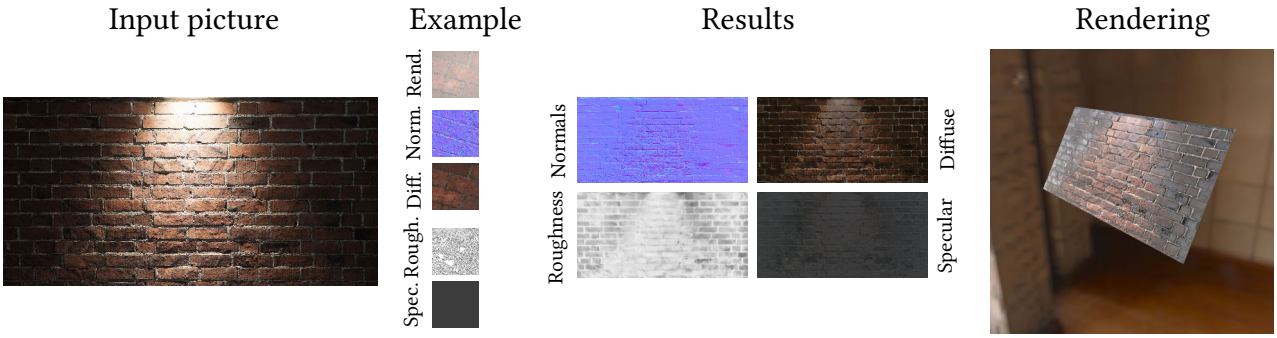


Figure 5.12: Limitation. Our method is not designed to handle large illumination gradients over the surface.

5.3 Conclusion

In this chapter we presented a method for large-scale material acquisition which allows more diversity and control in the capture. On one hand we provide user-control to the acquisition process through the example patches, allowing to influence properties of the material in the desired direction. On the other hand we allow material capture at arbitrary scale, resolution and aspect ratios, significantly improving the versatility in the capture process. In the future, we would like to generalize our approach to curved surfaces, possibly using some form of geometry reconstruction along with a suitable representation for deep learning, such as texture atlases. Furthermore, we hope that this work will inspire more research on deep material encoding for edition and acquisition.

Chapter 6

Industrial challenges

My PhD was funded by a collaboration between french "Agence Nationale Recherche Technologie" (ANRT) and Optis, an ANSYS affiliate. While my main responsibility was to lead research projects, it also included knowledge and technology transfer to Optis and exploration of solutions to industrial challenges related to my research.

6.1 Research transfer

The projects presented in this thesis were chosen to explore research directions close to Optis' interests. As challenges closer to industrial concerns arose, we pursued further experiments to adapt our methods. In this section I develop the main industrial concerns we encountered and how we tackled them. Some details are not included for confidentiality reasons.

6.1.1 Different model training

During our project we selected the Cook-Torrance model [CT82], provided in Substance Designer, for compatibility. This ensures that we render the materials as close from the original design as possible, for network training.

In comparison, Optis uses a material model, also inspired by Cook-Torrance, but using a distribution closer to the one described by Beckmann & Spizzichino [BS87]. This is a challenge, as the material dataset we have was not design to be rendered using Optis' model.

Fortunately, our rendering loss is agnostic to the material model since it compares the renderings of the training data and outputs, rather than material parameters. We therefore don't need the material models to be the same. We implement the output material model equation in our differentiable renderer, as illustrated in Figure 6.1. With this, we are able to train our network to output any material model able to represent the light behavior visible in the training data.

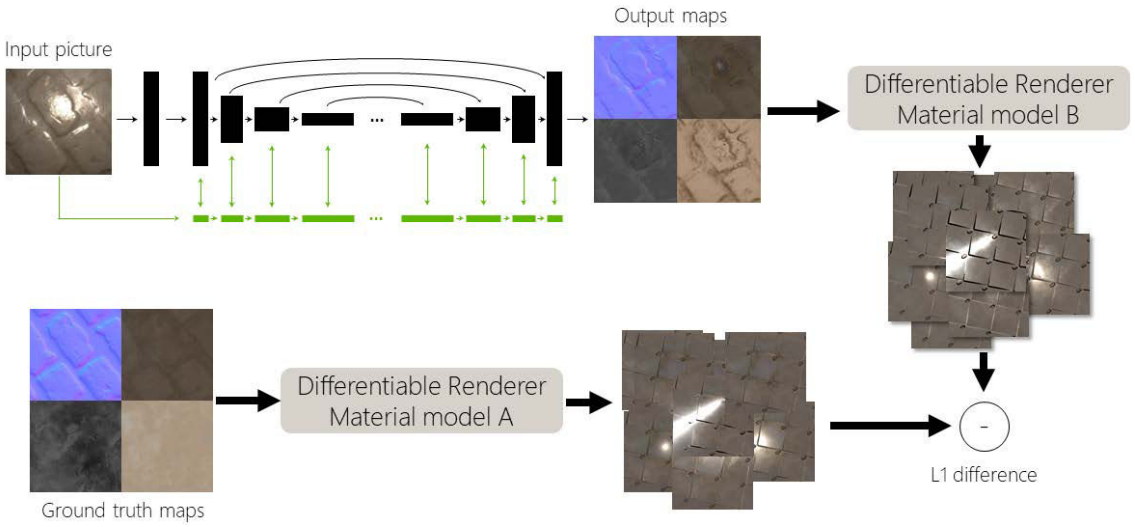
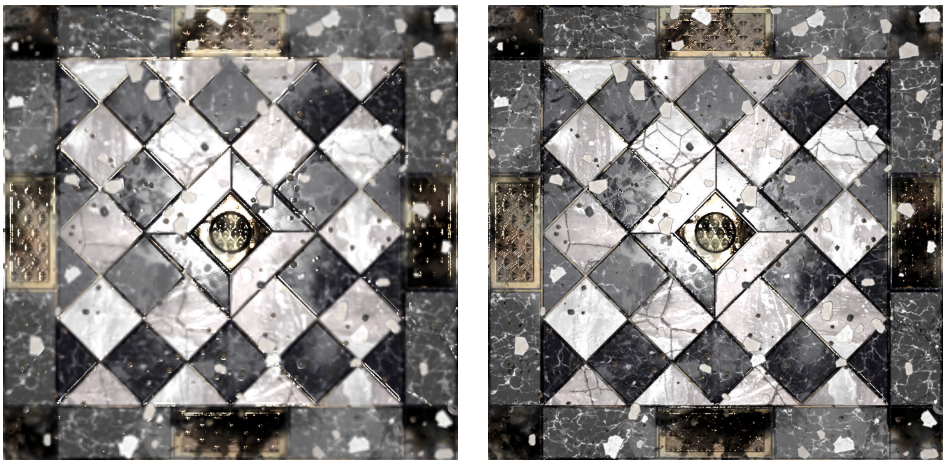


Figure 6.1: The rendering loss is agnostic to the material model used. We can train our network to output any material model able to represent all the effects displayed by the targets.

6.1.2 Resolution

A second limitation for industrial use is the definition of the output material maps. During our research projects described in Chapter 3 and Chapter 4, we trained the network with 256x256 images to increase the training speed and reduce the GPU memory consumption. A much higher resolution is required in industrial use of SVBRDFs to avoid the typical blurry appearance -visible in Figure 6.2. To solve this, we explored the possibility to extend our network with a few layers at the beginning of the encoder and the end of the decoder to use 1024x1024 inputs as shown in Figure 6.3. This approach seems to work well but requires long training time and large VRAM GPUs. In our experience, we trained this "HD" version for a week and see that despite it had not fully converged, the training behavior was similar. A concern would be that the training becomes less stable if we add too many layers. Karras et al. [KALL18] discuss a method to train GANs, which are famously unstable, for higher resolution. They propose to progressively increase the resolution during the training, by smoothly blending in new, higher resolutions, layers. I believe a similar technique could help maintain our network stable.

Another approach is to use the convolutional nature of the network, making it invariant



256x256 material 1024x1024 material

Figure 6.2: Images rendered on 1024x1024 resolution, using a 256x256 material on the left and a 1024x1024 resolution material on the right. This illustrates the blurriness created when the material resolution is too low compared to the rendering size.

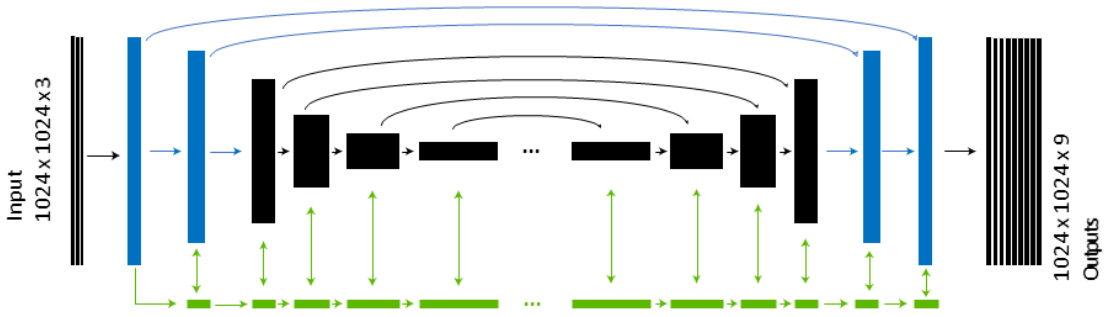


Figure 6.3: We illustrate here the architecture modification required to train a network with 1024x1024 images. In blue are the additional layers.

to the input size, as proposed by Gao et al.[GLD+19]. We present the result of this experiment in Figure 6.1.2. While a small resolution increase seem to maintain most of the network behavior, if we get too far from the original training size the quality goes down significantly. When increasing the size of the input, one also increases the size of the intermediate features, while the filters learned by the network do not adapt their size. Depending on the application, it may be difficult to maintain a good receptive field for the convolutions and a good encoding for the de-convolutions. We believe that this is the reason of the decreasing quality, visible in Figure 6.1.2

While these two options are interesting first approaches, we were still limited by the GPU memory and our use of close-up flash pictures for the resolution and scale of our acquisition. These limitations lead us to our work described in Chapter 5, removing the scale constraint, but also allowing for high resolution inference.

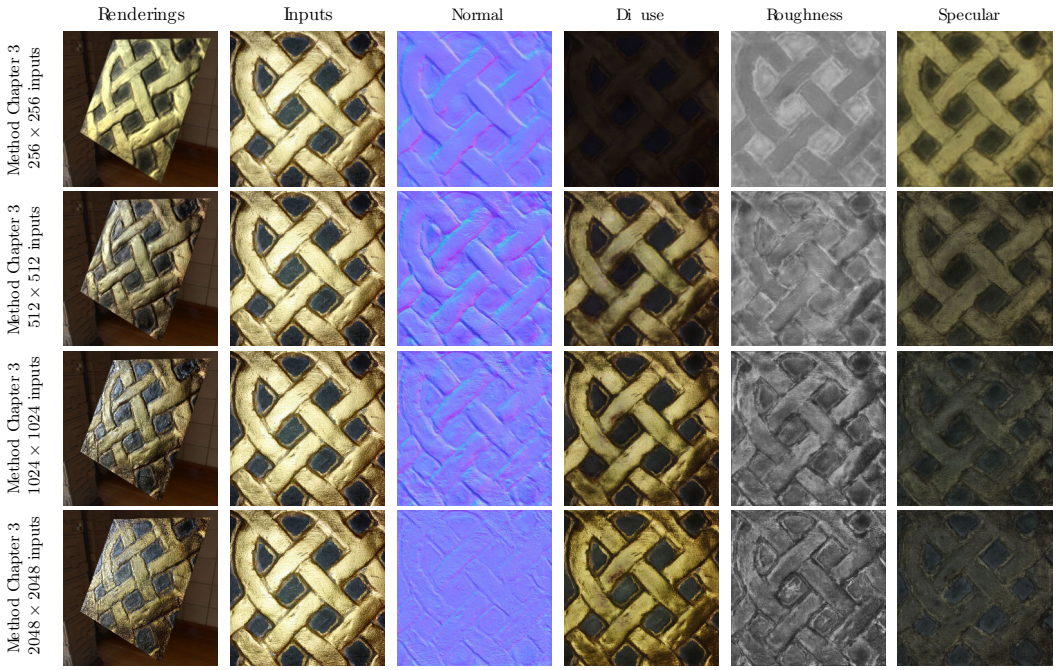


Figure 6.4: As our architecture is fully convolutional, the size of the input can be changed at inference time. Here we see the results of our one image method (described in chapter 3) using different sizes inputs. We can see that the results slowly degrades as the input resolution is further from the training size (256x256). This is visible in the normal which flattens, the metal specular behavior is slowly shifted to diffuse, the roughness gets noisy and the re-renderings are less faithful to the inputs with higher resolution. We believe it is due to the feature and cues size dramatical change and the internal features becoming too different for the decoder to interpret correctly.

6.2 Evaluation against gonioreflectometer

Our lightweight material acquisition methods produce plausible results, but without information about light and camera properties they are not correctly calibrated. While we evaluate our results in Chapter 3 and Chapter 4 against synthetic data ground truth, industrial users such as engineers and designers rely on physically correct behaviors of materials both for appearance and for optical property analysis. We therefore want to evaluate the error in BRDFs acquired through our algorithms against a specialized material acquisition hardware, and explore how to calibrate our results.

6.2.1 Acquisition process

For this evaluation method we use two acquisitions setups.

Gonioreflectometer The first setup is the Optis OMS2. It is a portable gonioreflectometer commercialized by Ansys, measuring isotropic homogeneous opaque materials in under a minute. The acquired material is measured using multiple incidence and view angles, producing a good quality tabulated BRDF representation. We consider the measurements of this device as ground truth for our experiment. On a perfectly diffuse sample, the accuracy and repeatability of the device are below 5% of error. The .brdf file that is produced, can be exported as conoscopic maps (see Figure 6.8), sampling, for a given wavelength, 5 incident angles for 400 x 400 theta-phi outgoing directions. We extract 3 conoscopic maps, corresponding to the RGB channels of usual analytic material models, respectively at 600nm, 550 nm, 450nm. I refer to this ground truth as the $BRDF_{measured}$.

Deep learning The second setup is our multi-image method, allowing to combine multiple pictures of a material from arbitrary light and view conditions to evaluate a Cook-Torrance SVBRDF model (described in Chapter 4). In our experiment, we combine 6 pictures taken with a single phone and its flash. Light and view direction are therefore almost collocated. Each picture is linearized and re-projected -as required by the method- before inference. As the $BRDF_{measured}$ is spatially homogeneous BRDFs, we remove the spatially varying component of the deep learning results by extracting an average of the values of the albedos for each material in the SVBRDF. We refer to these values in our experiment as the $BRDF_{inferred}$.

6.2.2 Error computation

To compute the error of the analytical model inferred by our multi-image method, we sample the material model function to generate a conoscopic map with the same representation as the $BRDF_{measured}$.

We define three errors. The ”Factor” is a float representing the average factor of multiplication between the inferred BRDF and the measured BRDF. Providing a ”global correction factor” for each wavelength.

$Factor = \frac{1}{N} \sum_{1}^{N} \frac{BRDF_{measured}}{BRDF_{Inferred}}$

N is the number of different light/view directions sampled in the conoscopic maps.

The "Absolute error" is defined by the absolute value of the difference between measured and inferred BRDFs

$Error_{Absolute} = \frac{1}{N} \sum_1^N |BRDF_{measured} - BRDF_{Inferred}|$

And finally, the "Relative error" calculation, gives us an estimation of how precise the $BRDF_{Inferred}$ is, relatively to the value of the $BRDFmeasured$.

$Error_{Relative} = \frac{1}{N} \sum_1^N \frac{|BRDF_{measured} - BRDF_{Inferred}|}{BRDF_{measured}}$

These three metrics provide us with a good understanding of the error in the inferred material, with both absolute error —$Error_{Absolute}$— representing the physical difference and relative errors —Factor and $Error_{Relative}$— putting the error in perspective.

6.2.3 Experiment

We design an experiment to evaluate the use of an average factor to "correct" the inferred BRDF. The diffuse and specular albedos are multiplied by this factor and we evaluate the new errors of this "corrected BRDF". This can be used to calibrate a spatially varying material with high frequency variation of the same material for example.

6.2.4 Results

I will describe the result on one of the acquired material and present the kind of visualization we use.

We use a X-Rite color checker to acquire sample materials as showed in Figure 6.5.

3D Projections

In Figure 6.6 & Figure 6.7 I evaluate with a 3D projection the measured BRDF (in blue) and the inferred BRDF (in red). I present here the results for 2 different incidences, before and after factor correction. We can see that the factor helps to correct most of the error caused by the unknown light power during acquisition, by correcting the albedos level, but does not influence the BRDF "shape".

Conoscopic maps

Conoscopic maps represent the amount of light reflected for a given light incidence for each outgoing θ and ϕ direction. Figure 6.8 & Figure 6.9 represent the conoscopic maps

Figure 6.5: X-rite Color checker. I will focus on the results for the blue BRDF.

and the errors of measured BRDF and inferred BRDF for a light incidence of 10°, before and after factor correction. We see that most of the error -in the right columns- is drastically reduced from over 200% to below 50% for this incidence angle.

BRDF slice

The figure 6.10 represents a slice of the BRDF with varying incident lighting (columns) for each RGB channel (lines). It is computed for $\phi_{Light} = \pi$ and $\phi_{View} = 0$. We found this representation to be a good summary of the BRDFs behavior. We can clearly see the effect of the factor correction, bringing the corrected BRDF (dotted red) significantly closer to the measured one (blue). We also evaluate the difference of using a single global factor for the whole BRDF (dotted red), or one different for each incidence (dotted green) and show that the difference is minor.

Finally, the Table 6.11 shows a summary of the quantitative errors.

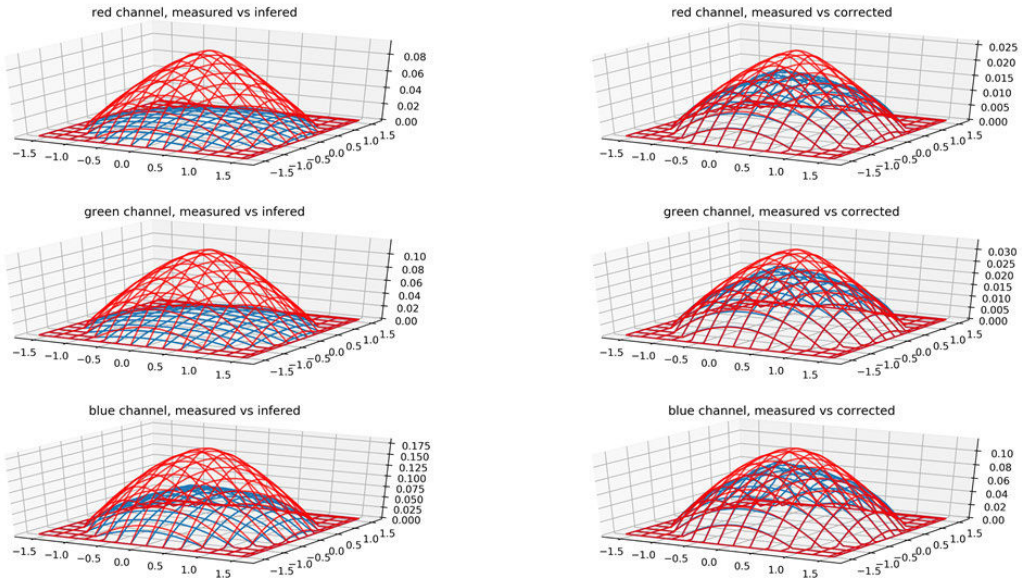


Figure 6.6: Blue BRDF intensity for 10° light incidence. Each row represents a color channel (R,G,B). In red is the inferred BRDF and in blue, the measured BRDF. The left column compares the intensities before correction and the right one after correction.

6.2.5 Conclusion

In this experiment, we show that a single inferred BRDF can be calibrated using a scaling, bringing the relative error below 20% in all our tests. Our hypothesis is that the scaling compensates for the unknown light power and white balance at acquisition time. An interesting direction is to explore how a multi-material SVBRDF acquired in a single picture set could be corrected and whether the correction is uniform. Future research directions could be to evaluate the effects of white balance on the result of our method or evaluate strongly specular materials such as metals.

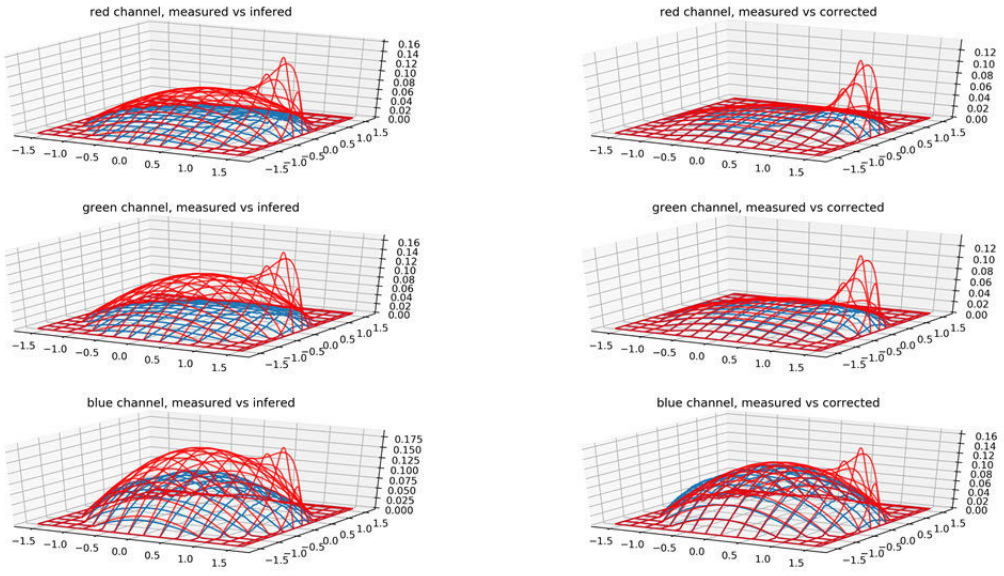


Figure 6.7: Blue BRDF intensity for 68.9° light incidence. Each row represents a color channel (R,G,B). In red is the inferred BRDF and in blue, the measured BRDF. The left column compares the intensities before correction and the right one after correction.

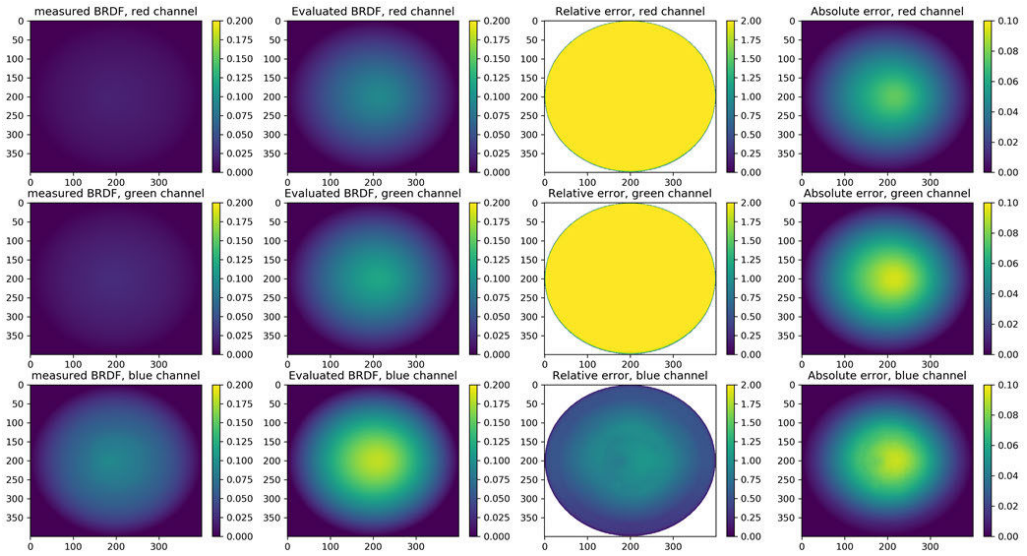


Figure 6.8: Column 1 and 2 represent the conoscopic map of the measured and inferred BRDF respectively for a light incidence of 10°. The two columns on the right are the relative error and absolute error between the measured and inferred BRDF.

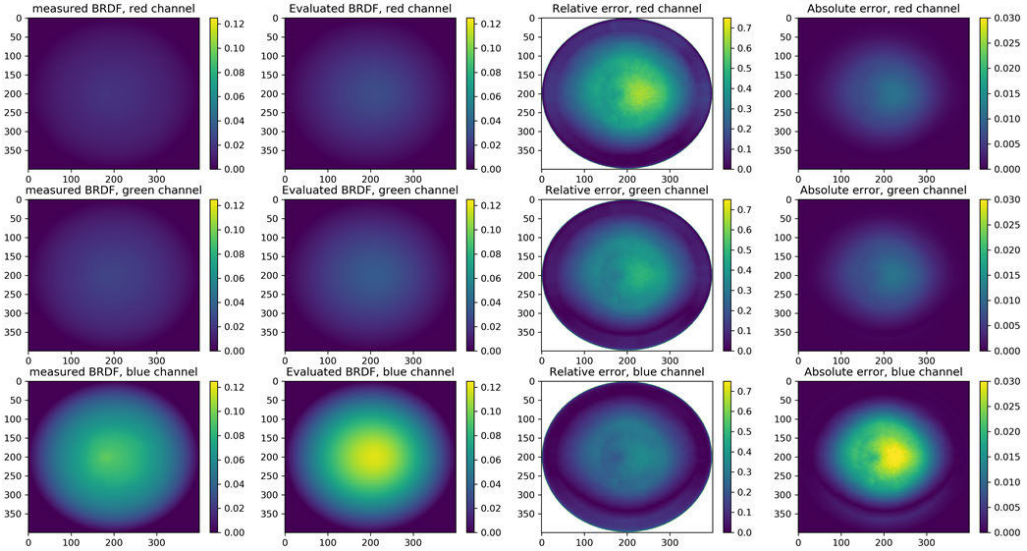


Figure 6.9: Column 1 and 2 represent the conoscopic map of the measured and corrected BRDF respectively for a light incidence of 10°. The two columns on the right are the relative error and absolute error between the measured and corrected BRDF

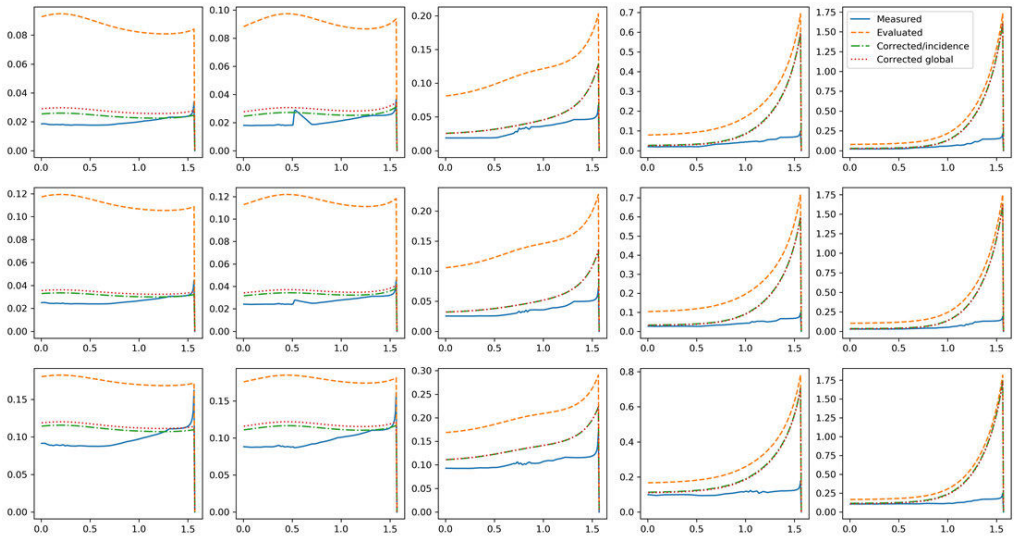


Figure 6.10: Cut in the blue BRDF for $\phi_{Light} = \pi$ and $\phi_{View} = 0$. Each column represents a different light incidence angle (10°, 22°, 44.9°, 59.9°, 68.9°). Each line represents a channel in R,G,B

Correction/Error Type	Factor	Relative Error	Absolute Error
Before correction	0.418	186%	0.0253
After correction	0.949	20%	0.0034

Figure 6.11: Summary of the different metrics introduced for the Blue BRDF.

Chapter 7

Conclusion

In this thesis, we showed how Computer Graphics and Deep Learning can be used together to design solutions to extract material information from pictures. We have presented a number of contributions to the lightweight material acquisition problem and will now summarize the main ones.

Data generation

When using supervised training, it can be difficult to gather a sufficient amount of good quality pairs of inputs and Ground Truth. We showed in Chapter 3 how Computer Graphics and domain specific knowledge allows to generate training data that is realistic enough to generalize well to real world examples. This approach not only allows to tackle complex problems for which large scale, precise labeled data is impossible to acquire, but also to easily adapt the dataset to various tasks. For example, once acquired, a real world dataset doesn't allow the modification of the camera or lighting conditions, while changing such parameters is trivial with synthetic data. While a classical data generation pipeline can be time consuming, we showed in Chapter 4 how data can be generated on the fly during the training process. This allows to quickly change the generation parameters and explore new tasks requiring different data, such as different acquisition setups in our context.

Loss design

In addition to the quantity and realism of our training data, the quality of our results stems from an approach that is aware of how the maps of our SVBRDF model interact together, thanks to our rendering loss presented in Chapter 3. With this contribution, we show that injecting problem specific knowledge in the optimization function is crucial to guide the training toward valid parts of the solution space. Furthermore, our rendering loss allows to easily change the material model our method infers, without changing the dataset ground-truths.

Architecture design

To further improve our results, we designed architectures capable of fusing distant information in the input image and to aggregate information available in an arbitrary number of input pictures. By specializing our network in Chapter 3, we take into account the specificity of our input data. Using a global feature track, we are able to extract valuable information from the flash in the picture, while compensating for the over-exposure in the photograph. In Chapter 4 we showed how to enhance the network to combine a variable, order independent, number of input images, allowing users to capture as many images as needed to exhibit all the visual effects of a material they want to capture. With this contribution, our method bridges the gap between single-image and many-image methods, allowing faithful material capture with a handful of images captured from uncalibrated light-view directions.

Acquisition scale, user control and high resolution

In Chapter 5 we alleviate inherent limitations of flash-based material acquisition methods, namely limited scale, low resolution, and lack of user control. By combining a guidance image with a few close-up exemplars, our approach can recover SVBRDFs of much larger surfaces, at high resolution and arbitrary aspect ratio. Furthermore, our design greatly increases the creative freedom of material designers by letting them create plausible SVBRDFs from existing photographs with fine control on their constituent materials.

Industrial environment

The industrial collaboration allowed me to tackle important challenges, described in Chapter 6 leading to the use of our methods in a production setup. I had the chance to discuss with artists using Optis' software and read feedback on material use in the industry. I was also able to access a gonioreflectometer, allowing me to work with complex acquisition tools and evaluate the challenges from a different perspective of material acquisition.

I will now discuss some research directions inspired by our results and the challenges encountered.

7.1 Future work

7.1.1 Similarity based on material deep features

As shown in Chapter 3, a meaningful metric is central to the success of training a deep network. We designed a loss to compare multiple renderings, taking the complex rendering pipeline into account. While this alleviates the problem of comparing different parametric models, the error is dominated by the diffuse albedo, as it impacts every single pixel of the rendering. Normal, roughness and specularity maps contribute to the total error less, as they have more spatially localized effects. The definition of a meaningful difference between materials remains an important challenge in the field.

I believe that new material representations would help to design a more representative perceptual distance. In the light of recent work on GANs and latent spaces [KLA18, ZPIE17], I am interested in exploring how deep learning can help define a more intuitive material representation by leveraging the internal layers of a specialized deep network. During training, these internal features are shaped into a compact encoding of the information most relevant to the optimization task, providing a new representation to work with and evaluate distance.

The interest I see in such deep features for materials goes beyond acquisition: a better perceptual representation would provide new leads for higher level edition, feature transfer — explored in Chapter 5 — and interpolation between materials. Recent work by Lagunas et al. [LMS+19] is an interesting approach to this problem, based on a massive user study encoded in a trained deep network.

7.1.2 Deep learning for material appearance

In this thesis we explore the use of standard deep network architectures as an optimization framework for material acquisition. Most network architectures are defined for general image processing tasks, and I believe that further specializing architectures to the material acquisition problem is essential. In Chapter 3, we show that adapting the loss and modifying the architecture to leverage material specific knowledge leads to significantly improved results.

In Chapter 4 we presented a method to aggregate information from multiple pictures. We can imagine that other sources of information can be provided to the network to match the quality of hundreds or thousands of calibrated pictures. A few examples of interesting information are white balance parameters, exposure levels, field of view or illuminance.

Another source of significant improvement in the same direction could be to leverage information from several complementary inputs. An example is the use of polarization filters [RRFG17] to explicitly separate specular from diffuse. Such intrinsic optical properties combined with deep learning would provide important input data. Their properties could be enforced in the loss function to guide the training, while maintaining a high level of acquisition convenience.

Further, a multi-spectral approach could bring complementary information. The near IR spectrum, for example, could provide cues about structure and material segmentation that can be lost due to over-exposure in a flash picture — as described in Chapter 3.

Exploiting additional sources of information could open deep learning methods to more complex material models, allowing to represent effects such as sub-scattering, transmission — important for faithful reproduction of materials like skin, glass — or cast shadows for better geometry capture, and help in calibrating results to physical quantities.

7.1.3 Complex acquisition as ground truth

So far, lightweight acquisition research mostly relies on artist-designed materials for training and lacks physically measured references to compare to. In Chapter 6, we propose a first step to solve this problem, but a more general approach is still required. For example, complex acquisition could be used to generate a high-quality dataset containing physically measured materials, fitted analytical models and multiple angle pictures: this would provide an important data set against which to test new methods.

7.1.4 Novel application domains

In the future, by combining the knowledge from Computer Graphics, Computer Vision, and Deep Learning, lightweight material acquisition methods could be improved to acquire multiple objects or even an entire room with a few pictures. Combined with a good geometry evaluation method, this would change the way virtual environments are designed and authored.

Two examples could be cultural heritage and medical examination of skin. Cultural heritage would strongly benefit from strong acquisition methods, designed to fit the particular constraints of the domain, in term of light exposure, fragility or accessibility of the artefacts for example. Finally, combined with more complex material representations and medical expertise, lightweight acquisition could lead the way to fast preliminary exams and treatment monitoring for medical applications such as skin lesion detection.

Appendix A

Appendix

A.1 Rendering loss pseudo-code

Algorithm 1 Rendering loss

Require: Ground Truth material(M_{gt}), Inferred material (M_{I}), Number of specular renderings (N_S), Number of diffuse renderings (N_D)

for n in N_S **do**

$P_{\text{S}} \leftarrow \text{Uniform}(-1.0, 1.0)$ ▷ Draw a position shift from a uniform distribution between -1 and 1

$\omega_{\text{V}} \leftarrow \text{Cosine}()$ ▷ Draw a 3D direction vector from the cosine distribution function

$P_{\text{V}} \leftarrow (\omega_{\text{V}} e^{\text{Normal}(\mu=0.5,\sigma^2=0.75)}) + P_{\text{S}}$ ▷ View position computation

$P_{\text{L}} \leftarrow (\omega_{\text{V}} * [-1, -1, 1]) e^{\text{Normal}(\mu=0.5,\sigma^2=0.75)}) + P_{\text{S}}$ ▷ Light position computation

$Ls \leftarrow \text{linspace}(-1, 1, 256)$

$C \leftarrow \text{concatenate}(\text{meshgrid}(Ls, Ls), \text{axis} = 2)$ ▷ Create a plane coordinate grid to calculate view/light direction

$\omega_{\text{V}} \leftarrow P_{\text{V}} - C$ ▷ Near-field view direction computation

$\omega_{\text{L}} \leftarrow P_{\text{L}} - C$ ▷ Near-field light direction computation

$\omega_{\text{LV}}.\text{add}([\omega_{\text{V}}, \omega_{\text{L}}])$ ▷ Store Light/View directions for future renderings

end for

for n in N_D **do**

$\omega_{\text{V}} \leftarrow \text{Cosine}()$ ▷ Distant view direction computation

$\omega_{\text{L}} \leftarrow \text{Cosine}()$ ▷ Distant light direction computation

$\omega_{\text{LV}}.store([\omega_{\text{V}}, \omega_{\text{L}}])$ ▷ Store Light/View directions for future renderings

end for

for $\omega_{\text{V}}, \omega_{\text{L}}$ in ω_{LV} **do**

$R_{\text{gt}}.\text{add}(\text{Render}(GT_M, \omega_{\text{V}}, \omega_{\text{L}}))$ ▷ Render the ground truths using computed directions

$R_{\text{I}}.\text{add}(\text{Render}(I_M, \omega_{\text{V}}, \omega_{\text{L}})$ ▷ Render the inferred materials using computed directions

end for

return $\text{mean}(|\log(R_{\text{gt}} + 0.01) - \log(R_{\text{I}} + 0.01)|)$ ▷ Compute the mean distance between all renderings

Algorithm 2 Cosine Random function

$r_1 \leftarrow \text{Uniform}(0.001, 0.95)$
$r_2 \leftarrow \text{Uniform}(0, 1)$
$r \leftarrow \sqrt{r_1}$
$\phi \leftarrow 2\pi r_2$
$x \leftarrow r\cos(\phi)$
$y \leftarrow r\sin(\phi)$
$z \leftarrow \sqrt{1.0 - r^2}$
return $[x, y, z]$

Algorithm 3 Rendering algorithm

Require: Incident lighting direction vectors (ω_L), View direction vector (ω_V), material $M =$ (Diffuse map (d), Normal map (n), Specular map (s), Roughness map (r)
Performed at each surface point separately
$h \leftarrow \text{Normalize}((\omega_\text{L} + \omega_\text{V})/2)$ ▷ Half-vector computation
$d \leftarrow \frac{d(1-s)}{\pi}$ ▷ Diffuse lighting computation
$D \leftarrow \frac{\frac{r^2}{(n.h)^2(r^4-1)+1}^2}{\pi_1}$ ▷ Cook-Torrance micro-facet normal distribution
$G \leftarrow \frac{1}{(n.\omega_\text{L})(1-\frac{r^2}{2})+\frac{r^2}{2}} \frac{1}{(n\cdot\omega_\text{V})(1-\frac{r^2}{2})+\frac{r^2}{2}}$ ▷ Cook-Torrance shadowing and masking term
$F \leftarrow s + (1-s)2^{((-5.55473(\omega_\text{V}\cdot h))-6.98316)(\omega_\text{V}\cdot h)}$ ▷ Fresnel effect approximation
return $\frac{(\frac{FGD}{4}+d)(n\cdot\omega_\text{L})}{\omega_\text{L}[3]}$ ▷ Final rendering equation compensated for low angles lighting directions with $\omega_\text{L}[3]$

Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AAL16] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 35(4), 2016.

[AD18] Miika Aittala and Fredo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, 2018.

[All19a] Allegorithmic. Substance designer, 2019.

[All19b] Allegorithmic. Substance share, 2019.

[AP07] Michael Ashikhmin and Simon Premoze. Distribution-based brdfs. Technical report, University of Utah, 2007.

[AWL13] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical SVBRDF capture in the frequency domain. 32(4), 2013.

[AWL15] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot SVBRDF capture for stationary materials. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 34(4):110:1–110:13, July 2015.

[BS87] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. 1987.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[CHW18] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. Ps-fcn: A flexible learning framework for photometric stereo. In *The European Conference on Computer Vision (ECCV)*, 2018.

[CK17] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

[CXG+16] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *IEEE European Conference on Computer Vision (ECCV)*, pages 628–644, 2016.

[DAD+18] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, 37(128):15, aug 2018.

[DAD+19] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Flexible svbrdf capture with a multi-image deep network. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 38(4), July 2019.

[DAW01] Ron O. Dror, Edward H. Adelson, and Alan S. Willsky. Recognition of surface reflectance properties from a single image under unknown real-world illumination. *Proc. IEEE Workshop on Identifying Objects Across Variations in Lighting: Psychophysics and Computation*, 2001.

[DBP+15] Olga Diamanti, Connelly Barnes, Sylvain Paris, Eli Shechtman, and Olga Sorkine-Hornung. Synthesis of complex image appearance from limited exemplars. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 34(2), 2015.

[DCP+14] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 33(6), 2014.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[DHT+00] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 145–156, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[dLDWS19] Riccardo de Lutio, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. Guided super-resolution as a learned pixel-to-pixel transformation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[Don19] Yue Dong. Deep appearance modeling: A survey. *Visual Informatics*, 3(2):59 – 68, 2019.

[DTPG11] Yue Dong, Xin Tong, Fabio Pellacini, and Baining Guo. Appgen: Interactive material modeling from a single image. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 30(6):146:1–146:10, 2011.

[DVGNK99] Kristin J Dana, Bram Van Ginneken, Shree K Nayar, and Jan J Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions On Graphics (TOG)*, 18(1):1–34, 1999.

[DWT+10] Yue Dong, Jinpeng Wang, Xin Tong, John Snyder, Moshe Ben-Ezra, Yanxiang Lan, and Baining Guo. Manifold bootstrapping for svbrdf capture. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4), 2010.

[FDA03] Roland W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3(5), 2003.

[FEW81] Bartell F., Dereniak E., and Wolfe W. The theory and measurement of bidirectional reflectance distribution function (brdf) and bidirectional transmittance distribution function (btdf). volume 0257, 1981.

[FJL+16] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. StyLit: Illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (proc. SIGGRAPH)*, 35(4), 2016.

[FJS+17] Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4), 2017.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[GCP+09] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul Debevec. Estimating specular roughness and anisotropy from second order spherical gradient illumination. *Computer Graphics Forum*, 28(4):1161–1170, 2009.

[GGG+16] Dar'ya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. BRDF Representation and Acquisition. *Computer Graphics Forum*, 2016.

[GLD+19] 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 Trans. Graph.*, 38(4):134:1–134:15, July 2019.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[GTHD03] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear light source reflectometry. *ACM Trans. Graph.*, 22(3):749–758, July 2003.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, and Tara Sainath. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, November 2012.

[HFM10] V. Havran, J. Filip, and K. Myszkowski. Bidirectional texture function compression based on multi-level vector quantization. *Computer Graphics Forum*, 29(1):175–190, 2010.

[HJO+01] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. *ACM SIGGRAPH*, 2001.

[HLT16] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European Conference on Computer Vision (ECCV)*, 2016.

[HLZ10] Michael Holroyd, Jason Lawrence, and Todd Zickler. A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 99:1–99:12, New York, NY, USA, 2010. ACM.

[HR76] Jack J. Hsia and Joseph C. Richmond. Bidirectional reflectometry. part i. a high resolution laser bidirectional reflectometer with results on several optical coatings. 1976.

[HSL+17] Z. Hui, K. Sunkavalli, J. Y. Lee, S. Hadap, J. Wang, and A. C. Sankaranarayanan. Reflectance capture using univariate sampling of brdfs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[IRWM17] C. Innamorati, T. Ritschel, T. Weyrich, and N. Mitra. Decomposing single images for layered photo retouching. *Computer Graphics Forum (Proc. EGSR)*, 36(4), 2017.

[ISSI16] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 35(4), 2016.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Jak10] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[Kaj86] James T. Kajiya. The rendering equation. In *Computer Graphics*, pages 143–150, 1986.

[KALL18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[KCW+18] Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. Efficient reflectance capture using an autoencoder. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 37(4), July 2018.

[KLA18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[KUMH17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 972–981. 2017.

[LBD+90] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[LCY+17] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2261–2269, 2017.

[LDPT17] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4), 2017.

[LKG+03] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2):234–257, 2003.

[LLM+18] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *CoRR*, abs/1807.03247, 2018.

[LMS+19] Manuel Lagunas, Sandra Malpica, Ana Serrano, Elena Garces, Diego Gutierrez, and Belen Masia. A similarity measure for material appearance. *ACM Transactions on Graphics (SIGGRAPH 2019)*, 38(4), 2019.

[LN16] Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38:129–141, 2016.

[LSC18] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. *Proceedings of ECCV*, 2018.

[LXR+18] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2018.

[Mca02] David Kirk Mcallister. *A Generalized Surface Appearance Representation for Computer Graphics*. PhD thesis, 2002.

[MCS90] J.F. Murray-Coleman and A.M. Smith. The automated measurement of brdfs and their application to luminaire modeling. *Journal of the Illuminating Engineering Society*, 19(1):87–99, 1990.

[MGSJW12] Francho Melendez, Mashhuda Glencross, Jonathan Starck, and Gregory J. Ward. Transfer of albedo and local depth variation to photo-textures. pages 40–48, 12 2012.

[MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[NMY15] Takuya Narihira, Michael Maire, and Stella X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[NRH+77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. Technical report, October 1977.

[PCF05] J. A. Paterson, D. Claus, and A. W. Fitzgibbon. Brdf and geometry capture from extended inhomogeneous samples using flash photography. *Computer Graphics Forum (Proc. Eurographics)*, 24(3):383–391, September 2005.

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, June 1975.

[QSMG16] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[QSMG17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Qui19] Quixel. Quixel, 2019.

[RGR+17] K. Rematas, S. Georgoulis, T. Ritschel, E. Gavves, M. Fritz, L. Van Gool, and T. Tuytelaars. Reflectance and natural illumination from single-material specular objects using deep learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[RJGW19] Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. Neural btf compression and interpolation. *Computer Graphics Forum (Proceedings of Eurographics)*, 38(2), March 2019.

[RK09] Roland Ruiters and Reinhard Klein. Btf compression via sparse tensor decomposition. *Computer Graphics Forum*, 28(4):1181–1188, 2009.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241, 2015.

[RPG16] J. Riviere, P. Peers, and A. Ghosh. Mobile surface reflectometry. *Computer Graphics Forum*, 35(1), 2016.

[RRFG17] Jérémy Riviere, Ilya Reshetouski, Luka Filipi, and Abhijeet Ghosh. Polarization imaging reflectometry in the wild. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2017.

[RVRK16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[RWS+11] Peiran Ren, Jinpeng Wang, John Snyder, Xin Tong, and Baining Guo. Pocket reflectometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), 2011.

[SQLG15] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[SSW+14] Christopher Schwartz, Ralf Sarlette, Michael Weinmann, Martin Rump, and Reinhard Klein. Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of bonn. *Sensors*, 14(5), April 2014.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces*. *J. Opt. Soc. Am.*, 57(9):1105–1114, Sep 1967.

[TZK+17] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[WAM02] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3), 2002.

[War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH*, 1992.

[WDR11] Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. A sparse parametric mixture model for btf compression, editing and rendering. *Computer Graphics Forum*, 30(2):465–473, 2011.

[WGGH18] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[WGK14] Michael Weinmann, Juergen Gall, and Reinhard Klein. Material classification based on training data synthesized using a btf database. In *European Conference on Computer Vision (ECCV)*, pages 156–171, 2014.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proc. of Eurographics Conference on Rendering Techniques (EGSR)*, 2007.

[WSM11] Chun-Po Wang, Noah Snavely, and Steve Marschner. Estimating dual-scale properties of glossy surfaces from step-edge lighting. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 30(6), 2011.

[WZ17] Olivia Wiles and Andrew Zisserman. Silnet : Single- and multi-view reconstruction by learning from silhouettes. *British Machine Vision Conference (BMVC)*, 2017.

[XNY+16] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal brdf sampling for two-shot near-field reflectance

acquisition. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016.

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*. 2017.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[ZREB06] T. Zickler, R. Ramamoorthi, S. Enrique, and P. N. Belhumeur. Reflectance sharing: predicting appearance from a sparse set of images of a known shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8), 2006.

[ZSQ+17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[ZSY+17] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas A. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[ZZI+17] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 9(4), 2017.